(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 981 223 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
23.02.2000 Bulletin 2000/08

(51) Int Cl.7: H04L 9/06

(21) Application number: 99306547.3

(22) Date of filing: 19.08.1999

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 20.08.1998 JP 23392198

(71) Applicant: KABUSHIKI KAISHA TOSHIBA
Kawasaki-shi, Kanagawa 210-8520 (JP)

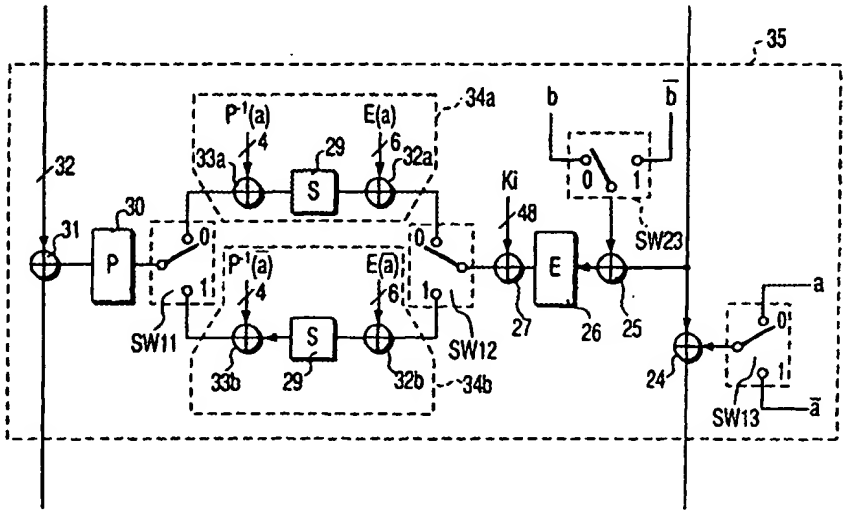
(72) Inventors:
- Kawamura, Shinichi,
IPD, Kabushiki Kaisha Toshiba
Tokyo 105-8001 (JP)
- Sano, Fumihiko, IPD, Kabushiki Kaisha Toshiba
Tokyo 105-8001 (JP)

(74) Representative: Shindler, Nigel
BATCHELLOR, KIRK & CO.
102-108 Clerkenwell Road
London EC1M 5SA (GB)

(54) Encryption/decryption apparatus

(57) A pair of a pattern of a mask (a) and a mask pattern obtained by bit inversion of the mask is prepared for each round function (5) in a data scrambler (1). Every time encryption is to be performed, one mask pattern of the pair is randomly selected by a switch (SW12), and an exclusive OR (32a) of an input to an S-box (29) and the selected mask pattern is calculated. In addition, an exclusive OR (33a) of an output from the S-box (29) and bits of inverse permutation p^{-1} of the mask (a) is calculated. The exclusive ORs (32a, 33a) are calculated in advance and stored as a table in the S-box (29). Furthermore, an exclusive OR (43a) of the output from each round function (5) and a mask (b) is calculated and concealed. The influence of the mask (b) is removed by calculating the exclusive OR with the mask (b) again on the next round.

FIG. 4



EP 0 981 223 A2

Description

[0001] This application is based on Japanese Patent Application No. 10-233921, filed August 20, 1998, the contents of which are incorporated herein by reference.

[0002] The present invention relates to an encryption/decryption apparatus and method and, more particularly, to an encryption/decryption apparatus and method which use secret key block encryption and a program storage medium therefor.

[0003] The DES (Data Encryption Standard) is secret key block encryption that has currently been used most widely, which is described in detail in Jpn. Pat. Appln. KOKAI Publication No. 51-108701.

[0004] The DES has been evaluated in various viewpoints, and decryption methods such as a differential decryption method and linear decryption method, which are more effective than a key exhaustive search method, have been proposed.

[0005] Note that the differential decryption method is disclosed in E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," Journal of CRYPTOLOGY, Vol. 4, Number 1, 1991. The linear decryption method is disclosed in Mitsuru Matsui, "Linear Decryption of DES ciphertext (I)", Encryption and Information Security Symposium, SCIS93-3C, 1993.

[0006] There is a new decryption method based on power consumption differences. In this method, power consumption differences between given bits of data (power consumption corresponding to bit 0 and power consumption corresponding to bit 1) are measured to estimate bits. In the case of the DES, for example, an input to an S-box and a corresponding output are estimated on the basis of a known ciphertext output and estimation of a key. A power consumption difference that appears when a given one bit is 0 or 1, which is estimated on the basis of the output from the S-box, is measured to check the validity of estimation, i.e., the validity of estimation of the key.

[0007] For this reason, there is a possibility that a DES ciphertext is decrypted by the above method, and hence higher security has been required.

[0008] It is an object of the present invention to provide an encryption/decryption apparatus and method which make it difficult to perform decryption by a technique based on power consumption differences without changing the data encryption processing result obtained by a conventional encryption/decryption apparatus and method, and a program storage medium for the apparatus and method.

[0009] In order to achieve the above object, according to the first aspect of the present invention, there is provided an encryption apparatus for converting a plaintext block into a ciphertext block depending on supplied key information, comprising means for randomly selecting one pattern of each of pairs ai, $\overline{ai}$ (where i is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time encryption is performed, means for masking bits dependent on a plaintext within the apparatus with the mask pattern selected by the selection means, and means for removing an influence of the mask a from a ciphertext before the ciphertext is output.

[0010] According to the second aspect of the present invention, there is provided an encryption apparatus for converting a plaintext block into a ciphertext block depending on supplied key information, comprising means for randomly selecting one pattern of each of pairs ai,$\overline{ai}$ (where i is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time encryption is performed, means for masking intermediate bit data within the apparatus with the mask pattern selected by the selection means, and means for removing an influence of the mask a from the intermediate bit data masked by the masking means.

[0011] According to the third aspect of the present invention, there is provided an encryption method of converting a plaintext block into a ciphertext block depending on supplied key information, comprising the steps of randomly selecting one pattern of each of pairs ai, $\overline{ai}$ (where i is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time encryption is performed, masking bits dependent on a plaintext within the method with the selected mask pattern, and removing an influence of the mask a from a ciphertext before the ciphertext is output.

[0012] According to the fourth aspect of the present invention, there is provided an encryption method of converting a plaintext block into a ciphertext block depending on supplied key information, comprising the steps of randomly selecting one pattern of each of pairs ai,$\overline{ai}$ (where i is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time encryption is performed, masking intermediate bit data within the method with the selected mask pattern, and removing an influence of the mask a from the masked intermediate bit data.

[0013] According to the fifth aspect of the present invention, there is provided a computer-usable program storage medium storing computer-readable program code means for converting a plaintext block into a ciphertext block depending on supplied key information, comprising computer-readable program code means for causing a computer to randomly select one pattern of each of pairs ai, $\overline{ai}$ (where i is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time encryption is performed, computer-readable program code means

for causing the computer to mask bits dependent on a plaintext within the method with the selected mask pattern, and computer-readable program code means for causing the computer to remove an influence of the mask $\underline{a}$ from a ciphertext before the ciphertext is output.

[0014] According to the present invention, original data is masked, and the mask is removed immediately before it is input to each S-box. When this mask is removed, there is a possibility that the data may be decrypted by the above technique based on power consumption differences. For this reason, according to the present invention, mask removal processing immediately before the data is input to each S-box, input operation of the original data to each S-box immediately after mask removal, and masking operation for the output from each S-box are calculated in advance and stored as a table, and the calculation result is obtained by looking up the table. For this reason, neither calculation of an exclusive OR for mask removal nor calculation of an exclusive OR for masking are performed during encryption and decryption, the data cannot be decrypted by the technique based on power consumption differences.

[0015] According to the present invention, consistency of encryption and decryption is ensured, and security against the decryption technique based on power consumption differences can be improved by making it difficult to decrypt data by the technique based on power consumption differences.

[0016] This summary of the invention does not necessarily describe all necessary features so that the invention may also be a sub-combination of these described features.

[0017] The invention can be more fully under stood from the following detailed description when taken in conjunction with the accompanying drawings, in which:

FIG. 1 is a block diagram showing the overall arrangement of a DES algorithm;
FIG. 2 is a block diagram showing the arrangement of the round functions of the DES;
FIG. 3 is a table showing an example of the contents of an S-box conforming to a DES standard table;
FIG. 4 is a block diagram showing an arrangement in which masks are added to round functions according to the present invention;
FIG. 5A is a circuit diagram showing an arrangement in which a mask is added to the input round according to the present invention;
FIG. 5B is a circuit diagram showing an arrangement in which a mask is added to the final round according to the present invention;
FIG. 6 is a table showing an expansion E;
FIG. 7 is a table showing a permutation P;
FIG. 8 is a view showing a concealed output from S1 which corresponds to an input (000000, 000001, ..., 111111) in the use of a mask $\underline{a}$;
FIG. 9 is a table of a mask $\bar{\underline{a}}$ (bit inversion of $\underline{a}$);
FIG. 10 is a block diagram showing an arrangement of a DES algorithm according to an embodiment;
FIG. 11 is a block diagram showing an arrangement obtained by adding masks to the round functions in the arrangement in FIG. 10;
FIG. 12 is a block diagram showing the arrangement of $\hat{S}$ in FIG. 11;
FIG. 13 is a block diagram showing another arrangement of a DES algorithm according to an embodiment;
FIG. 14 is a block diagram showing an arrangement obtained by adding masks to the round functions in the arrangement in FIG. 13;
FIG. 15 is a block diagram showing the arrangement of $\hat{S}$ in FIG. 14;
FIG. 16 is a block diagram showing the arrangement of a key scheduler of a DES algorithm;
FIG. 17 is a block diagram showing an arrangement in which a mask is added to the key scheduler according to the present invention;
FIG. 18 is a block diagram showing an arrangement in which the mask added to the key scheduler is added to each round function according to the present invention;
FIG. 19 is a flow chart showing the flow of processing in an encryption method according to an embodiment, which includes the step of masking bits dependent on a plaintext with selected mask patterns and the step of removing the influence of the masks described above from the ciphertext before it is output;
FIG. 20 is a flow chart showing the flow of processing in an encryption method according to an embodiment;
FIG. 21 is a flow chart showing the flow of processing in an encryption method according to an embodiment, which includes the step of removing the influence of masks from intermediate bit data during an encryption procedure and the step of masking the data with mask patterns;
FIG. 22 is a flow chart associated with an encryption procedure according to an embodiment of the present invention; and
FIG. 23 is a block diagram showing the arrangement of an IC card that implements the encryption/decryption method, and program storage medium therefor according to the present invention described above.

[0018] An embodiment of the present invention will be described below with reference to the views of the accompanying drawing.

[0019] FIG. 1 shows an arrangement of an encryption algorithm DES to which the present invention is applied. This arrangement is comprised of a data scrambler 1 including 1st to 16th rounds for scrambling a plaintext (64 bits) 3 depending on an externally input key 8 and outputting a corresponding ciphertext and a key scheduler 2 for expanding key information $\underline{k}$ into an interme-

diate key to be supplied to the data scrambler 1.

[0020] Referring to FIG. 1, the plaintext (64 bits) 3 is subjected to an initial permutation IP 4 first, and then divided into two equal halves. The left 32-bit data and right 32-bit data of the two equal halves are respectively input to a round function 5. The structure of the round function will be described later. The left 32-bit data and right 32-bit data output from the round function are interchanged and input to the next round function.

[0021] After these data are processed by the 16 round functions, a ciphertext 7 is output by a final permutation IP^{-1} 6.

[0022] FIG. 2 is a block diagram showing the details of the round function 5 in FIG. 1. A round function 17 is constituted by a permutation E 11, exclusive OR 13, S-boxes 14, permutation P 15, and exclusive OR 16.

[0023] The right 32-bit data is extended into 48-bit data by the permutation E 11. The resultant data is output to the exclusive OR 13. The exclusive OR 13 outputs the exclusive OR of the output from the permutation E 11 and an extended key 12. The 48-bit data output from the exclusive OR 13 is equally divided into 6-bit data. Each 6-bit data is input to a corresponding one of the S-boxes 14. In this embodiment, each S-box is formed from a table, and outputs 4-bit data with respect to a 64-entry 6-bit input. According to S1 based on the DES, if the left and right ends of a 6-bit input are respectively regarded as the first and sixth bits, a row in a table of the S-box in FIG. 3 is designated by the first and sixth bits regarded as binary numbers. Note that the row numbers in the table of the S-box shown in FIG. 3 are counted from above as the 0th, 1st, 2nd, and 3rd rows. A column number is then designated by the four remaining bits regarded as a binary number. The column numbers are also counted from the left end as the 0th 1st, 2nd, 3rd, ..., 15th columns. If, for example, 011011 is input to S1, the row number is 01. That is, the second row from above is designated. Since the column number is 01101, i.e., 13 (14th column from left), the value in the table is 15. Therefore, S1 outputs this value in binary notation, i.e., 0101. Referring to FIG. 3, each output from the S-box is designated by a row and column. In general, however, such an S-box is formed as a table corresponding to inputs ranging from 0 to 63. The 32-bit data obtained by combining outputs from the respective S-boxes is subjected to bit permutation operation by the permutation P 15. The resultant data is output to the exclusive OR 16. The exclusive OR 16 outputs the exclusive OR of the left 32-bit data and the output from the permutation P 15.

[0024] FIG. 4 is a circuit diagram showing the details of the round function 5 in FIG. 4 and the round function 17 in FIG. 2. FIG. 5A shows an arrangement for an input to the first round function. FIG. 5B shows an arrangement for an output from the 16th round function.

[0025] An embodiment of the present invention will be described in detail below with reference to FIGS. 4, 5A, and 5B.

[0026] Referring to FIG. 4, reference symbols $\underline{a}$ and $\underline{b}$ respectively denote 32-bit masks; and $\bar{a}$, inversion of all bits. In a round function 35 in FIG. 4, an exclusive OR 25 calculates the exclusive OR of the right 32-bit data and an output from a switch SW23 and outputs it to an expansion E 26. An output from the expansion E 26 is exclusive-ORed with an extended key Ki by an exclusive OR 27. The resultant data is output to a switch SW12. The switch SW12 causes the data to branch in accordance with a random number sequence Rij. If Rij is 0, the switch SW12 causes the data to branch to the 0 side. If Rij is 1, the switch SW12 causes the data to branch to the 1 side.

[0027] FIG. 4 shows the arrangement of each S-box after branching at the switch SW12. An S-box 29 corresponds to S1 to S8 based on the DES.

[0028] When the switch SW12 causes data to branch to the 0 side, the process indicated by a dashed line 34a is performed. More specifically, an exclusive OR 32a calculates the exclusive OR of the output from the exclusive OR 27 and six bits (E(a)) of the result obtained by performing the expansion E for the mask $\underline{a}$ which corresponds to an input of the S-box, and outputs the resultant data to the S-box 29. The S-box 29 outputs the result obtained by looking up the table of the S-box to an exclusive OR 33a.

[0029] The exclusive OR 33a calculates the exclusive OR of bits of $p^{-1}(a)$ as the result obtained by performing inverse permutation p^{-1} for the mask $\underline{a}$ and the output from the S-box 29, and outputs the resultant data to the switch SW11.

[0030] When the switch SW12 causes the data to branch to the 1 side, the process indicated by a dashed line 34b is performed. More specifically, an exclusive OR 32b calculates the exclusive OR of the output from the exclusive OR 27 and bits of the result obtained by performing the expansion E for the mask $\bar{a}$ which corresponds to an input of the S-box, and outputs the resultant data to the S-box 29. The S-box 29 looks up the table of the corresponding S-box and outputs the resultant data to the exclusive OR 33b.

[0031] The exclusive OR 33b calculates the exclusive OR of four bits of $p^{-1}(\bar{a})$ as the result obtained by performing inverse permutation p^{-1} of a permutation P(30) for the mask $\bar{a}$ which corresponds to an output from the S-box and the output from the S-box 29, and outputs the resultant data to the switch SW11.

[0032] Note that the processes indicated by the dashed lines 34a and 34b must not be performed during encryption and decryption. This is because, even if data is concealed with the above mask, since input/ output operation of the S-box 29 is not concealed, decryption may be attempted by using power consumption differences in S-box processing.

[0033] In this embodiment of the present invention, the results of the processes indicated by the dashed lines 34a and 34b are obtained first by pre-calculation performed before encryption and decryption, and en-

cryption processing and decryption processing are then performed. For example, a table in which the index of each input to each S-box and a corresponding output are rewritten is prepared for each S-box, and is used for encryption and decryption. In this case, a table of an S-box corresponding to the mask $\underline{a}$ and a table of an S-box corresponding to the mask $\bar{a}$ are prepared. For example, the following is the result obtained by calculating the process block 34a in FIG. 4 using the mask $\underline{a}$. Assume that the mask $\underline{a}$ is (0110 1111, 1100 1010 0110 1100 1100 0011). The expansion E is expressed by the table shown in FIG. 6. In the table shown in FIG. 6, the respective rows correspond to inputs to S1, S2, ..., S8 from above. In addition, the first bit on the left end of each column corresponds to the first bit of an input to a corresponding S-box. Each number in the table represents the Xth bit of a corresponding input to the expansion E. That is, the input to S1 includes the 32nd, 1st, 2nd, 3rd, 4th, and 5th bits of the input to E. With the above mask $\underline{a}$, therefore, a bit mask (a) corresponding to the input to S1 is (101101). FIG. 7 shows a table of the permutation P. Referring to FIG. 7, the numbers sequentially correspond to the first to 32nd bits of the output from P from the left (the first and second rows are contiguous). Each term represents the Xth bit of an input. That is, the first bit of the output from permutation P is the 16th bit of the input. The bits corresponding to S1 are the 1st, 2nd, 3rd, and 4th bits of the input to the permutation P, and hence respectively correspond to the 9th, 17th, 23rd, and 31st bits of the output from P. Since a mask corresponding to the output from S1 is p^{-1}(a), i.e., the output from P is $\underline{a}$, the 9th, 17th, 23rd, and 31st bits of the mask $\underline{a}$ become p^{-1}(a). The mask corresponding to the output from S1 is therefore (1001). With the above mask $\underline{a}$, therefore, a bit mask E(a) corresponding to the input to S1 is (101101), and a bit mask p^{-1}(a) corresponding to the output from S1 is (1001). According to the actually formed table corresponding to the mask $\underline{a}$, the output from S1 is calculated by using the result of the exclusive OR of the input and the bit mask E(a) as the input to S1, and an output from the table is obtained by adding the bit mask p^{-1}(a) to the output from S1 by exclusive OR. FIG. 8 shows an output (corresponding to an input of 0 to 63) of concealed S1 when the input corresponds to (000000, 000001, ..., 111111), in the case of the above mask $\underline{a}$. FIG. 9 shows a table of the mask $\bar{a}$ (bit inversion of $\underline{a}$).

[0034] The outputs from the respective process blocks indicated by the dashed lines 34a and 34b are permutated by a permutation P 30. The resultant data is output to an exclusive OR 31. The exclusive OR 31 calculates the exclusive OR of the left 32-bit data and the output from the permutation P 30. An exclusive OR 24 calculates the exclusive OR of the right 32-bit data and the output from the switch SW13 to obtain new right 32-bit data.

[0035] Referring to FIG. 5A, the result obtained by permutating the plaintext (64 bits) by initial permutation IP 41a is divided into equal halves, i.e., right 32-bit data and left 32-bit data. An exclusive OR 44a calculates the exclusive OR of the left 32-bit data and an output from a switch SW21. The output from this exclusive OR 44a becomes the left 32-bit data of an input of the first round function. An exclusive OR 42a calculates the exclusive OR of the right 32-bit data and an output from a switch SW14. An exclusive OR 43a calculates the exclusive OR of the output from the exclusive OR 42a and an output from a switch SW22.

[0036] The output from the exclusive OR 43a becomes the right 32-bit data of an input of the first round function. In the case shown in FIG. 5A, the sequence of the exclusive ORs 42a and 43a may be interchanged in accordance with the characteristics of the exclusive ORs.

[0037] Referring to FIG. 5B, the result obtained by permutating the plaintext (64 bits) by initial permutation IP 41a is divided into equal halves, i.e., right 32-bit data and left 32-bit data. An exclusive OR 44b calculates the exclusive OR of the left 32-bit data and the output from the switch SW21. This eliminates the influence of the mask in the exclusive OR 43a in FIG. 5A. The output from the exclusive OR 44b is input to a final permutation IP^{-1} 41b. An exclusive OR 42b calculates the exclusive OR of the right 32-bit data and the output from the switch SW14. An exclusive OR 43b calculates the exclusive OR of the output from the exclusive OR 42b and the output from switch SW22. This eliminates the influence of the mask in the exclusive OR 44a in FIG. 5A. The output from the exclusive OR 43b is input to the final permutation IP^{-1} 41b. Referring to FIG. 5B, the sequence of the exclusive ORs 42b and 43b may be interchanged in accordance with the characteristics of the exclusive ORs.

[0038] The characteristics of the arrangement shown in FIGS. 4, 5A, and 5B will be described below.

[0039] The exclusive ORs 44a, 42a, and 43a conceal data by using masks such as the masks $\underline{a}$ and $\underline{b}$. With this operation, in the data scrambler, it is difficult to observe the left 32-bit data and right 32-bit data from the outside world. If, however, data is concealed by using the above masks, inputs to the respective S-boxes 14 in FIG. 2 differ from the original plaintext data, and hence outputs from the S-boxes differ. Therefore, the output ciphertext does not correspond to the original plaintext data.

[0040] In order to solve the above problem, in each round function, the exclusive OR 25 in FIG. 4 calculates the exclusive OR of the mask $\underline{b}$ or mask $\bar{b}$. This eliminates the influence of concealment using the mask $\underline{b}$ or $\bar{b}$ added in FIG. 5A. If the switch SW12 causes the data to branch to the 0 side, the exclusive OR 32a eliminates the influence of concealment using the mask $\underline{a}$ in FIG. 5A. That is, the input to S29 becomes the same as the original plaintext input. The output from S29 is concealed again by the exclusive OR 33a using the mask $\underline{a}$. In this case, the process block 34a is performed in advance by looking up the table, no significant changes

in power consumption data directly associated with the input/output operation of S29 can be observed from the outside world.

[0041] The exclusive OR 24 in FIG. 4 temporarily eliminates the influence of the mask $\underline{a}$ or $\bar{a}$ on the right 32-bit data. However, the right 32-bit data is still concealed by the mask $\underline{b}$ or $\bar{b}$, and hence security is ensured. The right 32-bit data becomes left 32-bit data on the next round. The exclusive OR 31 calculates the exclusive OR of the left 32-bit data and the output from permutation P30. As a consequence, the data is concealed by the mask $\underline{a}$ (or $\bar{a}$) and mask $\underline{b}$ (or $\bar{b}$) and becomes a right input on the next round. As described above, therefore, consistency among the respective S-boxes is maintained in terms of DES for translation.

[0042] At the output of the final round, to eliminate the influence of each mask on concealment in FIG. 5A, the exclusive OR using each mask in FIG. 5B is performed.

[0043] The switches SW11, SW12, SW13, and SW14 are controlled by a random number sequence {R1i}. The switches SW21, SW22, and SW23 are controlled by a random number sequence {R2i}. For example, each switch selects a branch to the 0 side when Rji = 0, and selects a branch to the 1 side when Rji = 1. The random number sequences {R1i} and {R2i} for controlling the switches are characterized by being changed for each of encryption and decryption processes for the respective blocks. For example, in a given encryption process, all the switches SW11, SW12, SW13, and SW14 on the respective rounds perform processing on the 0 side. In another encryption process, all the switches SW11, SW12, SW13, and SW14 on the respective round perform processing on the 1 side.

[0044] If there is a clear relationship of dependence between the random number sequences {R1i} and {R2i}, an attacker has a clue to the estimation of the masks $\underline{a}$ and $\underline{b}$, random number sequences having no clear relationship of dependence are used as the random number sequences {R1i} and {R2i}. Ideally, the use of two random number sequences which are statistically independent is recommended. In practice, however, even if there is a statistical dependence relationship, this technique is effective as a measure against decryption based on power consumption differences, as long as the influence is sufficiently small. Two $\underline{m}$ sequence generators may be prepared as means for implementing the present invention, and outputs from the first and second m sequence generators may be respectively set to {R1j} and {R2j}. If the period of an $\underline{m}$ sequence is sufficiently long and the sequence lengths of the two m sequence generators, corresponding convention polynomials, and part or all of initial values are made to differ from each other, the above condition can be sufficiently satisfied. As another means for implementing random number sequences, one $\underline{m}$ sequence generator may be prepared to generate two bits for each encryption or decryption process. The first and second bits are respectively used as {R1j} and {R2j}.

[0045] Although the $\underline{m}$ sequence generators are presented as practical examples in this case, any random number sequence generator can be used as long as security in practice can be ensured. Note that these random number sequences must be implemented so as not to be estimated from the outside world. According to still another implementation means, random number sequences may be stored in a memory in advance to be sequentially referred to. Note that these random number sequences must be implemented so as not to be estimated from the outside world.

[0046] Referring to FIGS. 4, 5A, and 5B, the number of 1s of a bit sequence, i.e., a Hamming weight, is defined as H(a). In decryption using the technique based on power consumption differences, power consumption differences in a data encryption process are observed to acquire information about an encryption key. The concealment of data using the above masks makes it difficult to bring power consumption measurement from the outside world into correspondence with processed data. If, however, the Hamming weights of masks differ from each other, only data using only the masks $\underline{a}$ and $\underline{b}$ may be extracted in accordance with measurement of a plurality of encryption data and statistical information. If, only such data can be extracted, a key can be extracted as in the prior art by using the decryption technique based on power consumption differences. Since the currently used mask can be discriminated as the mask $\underline{a}$ or $\bar{a}$ in this manner, satisfactory countermeasures cannot be taken. If, for example, the Hamming weights of the masks $\underline{a}$ and $\bar{a}$ or masks $\underline{b}$ and $\bar{b}$ are set to be equal, it is difficult to discriminate the masks by measurement from the outside world, thus ensuring security. If, however, the bit weights of the masks are offset, the security greatly deteriorates.

[0047] Referring to FIGS. 4, 5A, and 5B, if, therefore, masks that satisfy $H(a) = H(\bar{a}) = H(b) = H(\bar{b}) = n/2 = 16$ are selected (the Hamming weights of the masks are equal to each other), high security is ensured. In this case, since a bit count $\underline{n}$ of each of the masks $\underline{a}$ and $\underline{b}$ is 32, a mask value of 16 is preferably used as the bit weight of each of the masks $\underline{a}$ and $\underline{b}$ and the bit inversions of the masks $\underline{a}$ and $\underline{b}$. Ideally, as described above, a mask having a Hamming weight corresponding to half of the bit length of the mask is preferably used. However, the same effect as described above can be obtained by using two masks having almost the same Hamming weight. In other words, if the Hamming weight indicating the number of bits 1 of an n-bit long bit sequence $\underline{x}$ is defined as H(x), the Hamming weight H(a) of the mask $\underline{a}$ satisfies $0 < H(a) < n$. Alternatively, the absolute value of the difference between the Hamming weight H(a) of the mask $\underline{a}$ and the Hamming weight $H(\bar{a})$ of the bit inversion $\bar{a}$ of the mask $\underline{a}$ is less than n/2.

[0048] That is, if the Hamming weights of the respective masks are not extremely offset, it is not easy to discriminate the masks by measurement from the outside world. Therefore, the effect of a countermeasure against

the technique based on power consumption differences can be obtained.

[0049] Consider the characteristics of the expansion E 26 based on the DES in FIG. 4. For the same reason as that for the selection of a mask value in consideration of Hamming weights, masks whose Hamming weights E(a) and E($\bar{a}$) applied to the exclusive ORs 32a and 32b are equal to each other are selected. That is, masks satisfying H(E(a)) = H(E($\bar{a}$)) are selected.

[0050] When the above mask condition is applied to the implementation of the DES, for example, it is required that both the number of 1s of the "first bits" (the bits on the left ends) of the respective 4-bit blocks of the mask $\underline{a}$ and the number of 1s of the "fourth bits" (the bits on the right ends) of the respective 4-bit blocks of the mask $\underline{a}$ are 4 each. That is, this embodiment is characterized by selecting the masks $\underline{a}$ and $\underline{b}$ that satisfy the above condition. As mask value that satisfy the above condition,

$(1000001111101101111001010010000 1)_2$,

$(1101101001100101001101011000 1010)_2$, and the like can be used.

[0051] Ideally, the use of mask values that satisfy the above condition is recommended. However, a similar effect can be obtained if "the number of 1s of the "first bits" of the respective 4-bit blocks of the mask $\underline{a}$" and "the number of 1s of the "fourth bits" of the respective 4-bit blocks of the mask $\underline{a}$" are not extremely offset.

[0052] In using the mask values that satisfy the above condition, when there is no clear correspondence between the random number sequences {R1j} and {R2j} for controlling the switches, even if the same mask value is used for the masks $\underline{a}$ and $\underline{b}$, effective countermeasures can be taken against decryption using the technique based on power consumption differences.

[0053] The DES arrangement shown in FIG. 1 is most widely known. However, DES arrangement methods having undergone various equivalent modifications to attain an increase in processing speed have been known.

[0054] Modifications in which the present invention is applied to the DES will be described below.

[0055] FIG. 10 shows an equivalent modification of the DES. In the implementation of the DES in FIG. 10, in order to improve the processing efficiency, the permutation E 11 and the permutation P 15 are integrated into one permutation and processed as an EP 53. The output obtained by permutating an input plaintext 58 by an initial permutation IP 57 is divided into equal halves. The right 32-bit data is input to an expansion E 51a, and the left 32-bit data is input to an expansion E 51b. The 48 bits output from the expansion E 51a are the right 48 bits of an input to the first round. The 48 bits output from the expansion E 51b are the left 48 bits of an input to the first round. An exclusive OR 55 calculates the exclusive OR of the right 48 bits of the input and an extended key K1, and outputs the resultant data to an S-box 54. The S-box 54 outputs a corresponding output to the EP 53 by looking up the table. The EP 53 permutates the input and outputs the resultant data to an exclusive OR 56. The exclusive OR 56 calculates the exclusive OR of the left 48 bits output from the expansion E 51a and the output from the EP 53. The resultant data becomes the right 48 bits of an input to the next round. The above processing on the first round is repeated up to the 16th round. The right 48 bits output from the 16th round are input to a contraction permutation E^{-1} 52a, and the left 48 bits are input to a contraction permutation E^{-1} 52b. The respective 32-bit outputs are input to a final permutation IP^{-1} 59. As a consequence, a 64-bit ciphertext 60 is output.

[0056] A method of preventing decryption using the technique based on power consumption differences by applying the present invention to such a modified DES will be described below.

[0057] FIG. 11 shows an embodiment of the implementation of the DES in FIG. 10 according to the present invention. Referring to FIG. 11, "E(a)/E($\bar{a}$)" indicates how the switch SW23 applies a mask based on an exclusive OR. That is, "E(a)/E($\bar{a}$)" indicates the mask E(a) or E($\bar{a}$).

[0058] FIG. 11 shows an embodiment which indicates that the present invention shown in FIGS. 4, 5A, and 5B can be applied to the implementation of the DES in FIG. 10.

[0059] The output obtained by performing an initial permutation for an input plaintext is divided into two equal halves. The right 32-bit data is input to an expansion E 61a, and the left 32-bit data is input to an expansion E 61b. An exclusive OR 64 calculates the exclusive OR of the 48-bit data output from the expansion E 61a and the mask E(a)/E($\bar{a}$) and outputs the resultant data to an exclusive OR 65. The exclusive OR 65 calculates the exclusive OR of the output from the exclusive OR 64 and the mask E(b)/E($\bar{b}$) to obtain the right 48 bits of an input to the first round. Note that the sequence of the exclusive ORs 64 and 65 may be interchanged depending on the characteristics of the exclusive ORs.

[0060] An exclusive OR 69 calculates the exclusive OR of the 48-bit data output from the expansion E 61b and the mask E(b)/E($\bar{b}$) to obtain the left 48 bits of an input to the first round.

[0061] An exclusive OR 66 calculates the exclusive OR of the right 48 bits of the input and the mask E(a)/E($\bar{a}$) to obtain the left 48 bits of an input to the next round. An exclusive OR 67 calculates the exclusive OR of the right 48 bits of the input and the E(b)/E($\bar{b}$) and outputs the resultant data to an exclusive OR 68. The exclusive OR 68 calculates the exclusive OR of the output from the exclusive OR 67 and the extended key K1 and outputs the resultant data to $\hat{S}$ 62 ("^" indicates exponentiation). The structure of $\hat{S}$ 62 will be described later. The output from $\hat{S}$ 62 is permuted by an EP 63 and output to an exclusive OR 70.

[0062] The shift register 70 calculates the exclusive OR of the left 48 bits of the input data and the output

from the EP 63 to obtain the right 48 bits of an input to the next round. The processing on the first round is repeated up to the 16th round. The output from the final round is subjected to processing reverse to that for the input to the first round. More specifically, the right 48 bits are subjected to the exclusive OR 65, exclusive OR 64, and contraction permutation E^{-1}, whereas the left 48 bits are subjected to the exclusive OR 65 and contraction permutation E^{-1}. The resultant two 32-bit data are output to the final permutation.

[0063] FIG. 12 shows the structure of $\hat{S}$ 62 in FIG. 11.

[0064] Referring to FIG. 12, $\alpha = E(a)$ and $\bar{\alpha} = E(\bar{a})$. An exclusive OR 71 calculates the exclusive OR of an input to $\hat{S}$ 62 and α mask $\bar{\alpha}$ or a and inputs the resultant data to an S-box 72. An exclusive OR 73 calculates the exclusive OR of the output from the S-box 72 and a mask $p^{-1}E^{-1}(\alpha)$ or $p^{-1}E^{-1}(\bar{\alpha})$ to obtain an output from $\hat{S}$ 62.

[0065] That is, a block 74 in FIG. 12 corresponds to the process blocks 34a and 34b including the switches SW12 and SW11 in FIG. 4. Note, however, that the process in the block 74 must not be performed during encryption and decryption. This is because, even if data is concealed with the above mask, since input/output operation of the S-box 72 is not concealed, decryption may be attempted by using power consumption differences in S-box processing.

[0066] The embodiment of the present invention is characterized in that the result of the process in the block 74 is obtained first by calculation performed in advance before encryption and decryption, and are then used for encryption processing and decryption processing. For example, a table in which the index of each input to each S-box and a corresponding output are rewritten is prepared for each S-box and used as $\hat{S}$ for encryption processing and decryption processing. In this case, an $\hat{S}$ table corresponding to the mask α and an $\hat{S}$ table corresponding to the mask $\bar{\alpha}$ are prepared in each S-box.

[0067] FIG. 13 shows another equivalent modification of the DES.

[0068] In the implementation of the DES in FIG. 13, in order to improve the processing efficiency, the expansion E 11 and permutation P 15 are integrated into one permutation and processed as an EP 83. The output obtained by permutating an input plaintext 88 by an initial permutation IP 87 is divided into two equal halves. The right 32-bit data is input to a permutation p^{-1} 81a, and the left 32-bit data is input to a permutation p^{-1} 81b. The 32 bits output from the permutation p^{-1} 81b are the right 32 bits of an input to the first round. The 32 bits output from the permutation p^{-1} 81b are the left 32 bits of an input to the first round. The right 32 bits of the input are input to the EP 83, and the resultant data obtained by performing an expansion for the input is output to an exclusive OR 85. The excitation reconstruction section 85 calculates the exclusive OR of the data and the extended key K1 and outputs the resultant data to an S-box 84. The S-box 84 outputs a corresponding output to an exclusive OR 86 by looking up the table. The exclusive OR 86 calculates the exclusive OR of the left 32 bits output from the expansion E 81b and the output from the S-box 84 to obtain the right 32 bits of an input to the next round. The processing on the first state is repeated up to the 16th round.

[0069] At the output of the 16th state, the right 32 bits are input a permutation P 82a, and the left 32 bits are input to a permutation P 82b. The respective 32-bit data are input to a final permutation IP^{-1} 89. As a consequence, a 64-bit ciphertext 90 is output. A method of preventing decryption using the technique based on power consumption differences by applying the present invention to such a modification of the DES will be described below.

[0070] FIG. 14 shows an embodiment of the equivalent modification of the DES in FIG. 13 according to the present invention.

[0071] Referring to FIG. 14, "$p^{-1}(a)/p^{-1}(\bar{a})$" indicates how the switch SW23 applies a mask based on an exclusive OR. That is, "$p^{-1}(a)/p^{-1}(\bar{a})$" indicates a mask $p^{-1}(a)$ or $p^{-1}(\bar{a})$.

[0072] FIG. 14 shows an embodiment which indicates that the present invention shown in FIGS. 4, 5A, and 5B can be applied to the implementation of the DES in FIG. 13.

[0073] The output obtained by performing an initial permutation for an input plaintext is divided into two equal halves. The right side 32-bit data is input to a permutation p^{-1} 91a, and the left 32-bit data is input to a permutation p^{-1} 91b. An exclusive OR 94 calculates the exclusive OR of the 32 bits output from the permutation p^{-1} 91a and $p^{-1}(a)/p^{-1}(\bar{a})$ and outputs the resultant data to an exclusive OR 95. The inverter circuit 95 calculates the exclusive OR of the output from the exclusive OR 94 and the mask $p^{-1}(a)/p^{-1}(\bar{a})$ to obtain the right 32 bits of an input to the first round. Note that the sequence of the exclusive ORs 94 and 95 may be interchanged depending on the characteristics of the exclusive ORs.

[0074] An exclusive OR 96 calculates the exclusive OR of the right 32 bits of the input and the mask $p^{-1}(a)/p^{-1}(\bar{a})$ to obtain the left 34 bits of an input to the next round. An exclusive OR 97 calculates the exclusive OR of the right 32 bits of the input and the mask $p^{-1}(b)/p^{-1}(\bar{b})$ and outputs the resultant data to an EP 93. The 48-bit output obtained by expansion at the EP 93 is output to an exclusive OR 98 to be exclusive-ORed with the enlarge key K1. The resultant data is output to $\hat{S}$ 92. The structure of $\hat{S}$ 92 will be described later. The output from $\hat{S}$ 92 is output to an exclusive OR 100 to be exclusive-ORed with the left 32 bits of the input data so as to obtain the right 32 bits of an input to the next round. The above processing on the first state is repeated up to the 16th round.

[0075] The output from the final round is subjected to processing reverse to that for the input to the first round. More specifically, the right 32 bits are subjected to the exclusive OR 95, exclusive OR 94, and permutation P,

whereas the left 32 bits are subjected to the exclusive OR 95 and permutation P. The resultant two 32-bit data are output to the final permutation.

[0076] FIG. 15 shows the structure of $\hat{S}$ 92 in FIG. 14.

[0077] Referring to FIG. 15, $\bar{\alpha} = p^{-1}(a)$ and $\bar{\alpha} = p^{-1}(\bar{a})$. An exclusive OR 101 calculates the exclusive OR of an input to $\hat{S}$ 92 and a mask α or $\bar{\alpha}$ and inputs the resultant data to an S-box 102.

[0078] An exclusive OR 103 calculates the exclusive OR of the output from the S-box 102 and a mask $p^{-1}E^{-1}(\alpha)$ or $p^{-1}E^{-1}(\bar{\alpha})$ to obtain an output from $\hat{S}$ 92. That is, a block 104 in FIG. 15 corresponds to the process blocks 34a and 34b including the switches SW12 and SW11 in FIG. 4. Note, however, that the process in the block 104 must not be performed during encryption and decryption. This is because, even if data is concealed with the above mask, since input/output operation of the S-box 102 is not concealed, decryption may be attempted by using power consumption differences in S-box processing. The embodiment of the present invention is characterized in that the result of the process in the block 104 is obtained first by calculation performed in advance before encryption and decryption, and are then used for encryption processing and decryption processing. For example, a table in which the index of each input to each S-box and a corresponding output are rewritten is prepared for each S-box and used as $\hat{S}$ for encryption processing and decryption processing.

[0079] In this case, an $\hat{S}$ table corresponding to the mask α and an $\hat{S}$ table corresponding to the mask $\bar{\alpha}$ are prepared in each S-box.

[0080] An embodiment in which the present invention is applied to a key scheduler will be described next with reference FIGS. 16, 17, and 18.

[0081] A mask pattern $\underline{c}$ for masking a bit pattern K of a true key and a bit inversion pattern $\bar{c}$ are prepared. Let Kc be the value obtained by converting K with $\underline{c}$ by using designated dyadic operation, and $K\bar{c}$ be the value obtained by converting K with $\bar{c}$ by using the same dyadic operation. The values Kc and $K\bar{c}$ are stored in the memory in advance. Every time encryption or decryption is executed, one of the values Kc and $K\bar{c}$ is randomly selected and processed in the same manner as the true key. The resultant data is applied to a plaintext by the above dyadic operation, and inversion of the dyadic operation is performed to remove the influence of the pattern $\underline{c}$ or $\bar{c}$ from the output obtained by the dyadic operation. A case wherein the present invention is applied to a DES scheme as an encryption scheme using exclusive OR operation as dyadic operation will be described first. First of all, two masked keys Kc and $K\bar{c}$ are prepared:

$$Kc = K(+)c$$

$$K\bar{c} = K(+)\bar{c}$$

where (+) represents an exclusive OR for each bit.

[0082] Prior to encryption or decryption, one of the keys Kc and $K\bar{c}$ is randomly selected, and a key schedule process of the DES is performed to sequentially generate 16 extended keys. The 16 keys extended from Kc are expressed by Kci (i = 1, ..., 16), and the keys extended from $K\bar{c}$ are expressed by $K\bar{c}i$ (i = 1, ..., 16). The keys extended from Kc are influenced by the mask $\underline{c}$, and the keys extended from $K\bar{c}$ are influenced by the mask $\bar{c}$. This influence is determined by the key schedule process of the DES. In this case, however, the keys extended from the true key K, which is not masked, according to a key schedule are expressed by Ki (i = 1, ..., 16), the exclusive OR of Ki and Kci is expressed by ci, and the exclusive OR of Ki and $K\bar{c}i$ is expressed by $\bar{c}i$. That is, $ci = Ki(+)Kci$ $\bar{c}i = Ki(+)K\bar{c}i$

[0083] In the DES, each extended key Ki is applied to a message by an exclusive OR for each bit immediately after the expansion E. In the present invention, Kci or $K\bar{c}i$ is applied in place of Ki. When $K\bar{c}i$ is applied, its influence is removed by applying $\bar{c}i$ by exclusive OR operation after the application of Kci. When $K\bar{c}i$ is applied, its influence is removed by applying ci by exclusive OR operation after the application of Kci. The values ci and $\bar{c}i$ are obtained by enlarging $\underline{c}$ and $\bar{c}$ according to the key schedule of the DES in the same manner as extended keys. The value ci or $\bar{c}i$ may be generated from the mask $\underline{c}$ or $\bar{c}$ selected every time encryption or decryption is performed. However, the method of calculating ci and ci in advance is the method that can suppress the leakage of information most against observation from the outside world. In this case, two sets of 16 48-bit masks, i.e., a total of 1,536 bits, are prepared. When, for example, the present invention is applied to IC cards, since these masks can be fixed at least for each card, ci and $\bar{c}i$ can be written in the ROM. This is important especially for IC cards whose storage capacities are severely constrained. In general, when the same number of bits are to be stored, the area of a ROM is smaller than that of a RAM or EEPROM. When a 1,536-bit mask is stored in a ROM, the use efficiency of an LSI chip area becomes higher than when the mask is stored in a RAM or EEPROM.

[0084] FIG. 16 shows a key schedule of the DES.

[0085] Referring to FIG. 16, reference symbols (PC - 1) 111 and (PC - 2) 113 denote functions each constituted by a combination of bit selection and a permutation; and ROT 112, cyclic shift operation. (PC - 1) 111 discards eight bits of an externally input 64-bit key K 115 and transfers two 28-bit sequences to the cyclic shift 112. The cyclically shifted data consisting of a total of 56 bits is input to (PC - 2) 113 to output a 48-bit extended key. Referring to FIG. 16, only the extended key corresponding to one round is output. However, extended keys corresponding to the 2nd, 3rd, ..., 16th rounds are generated by repeating the cyclic shift and PC - 2.

[0086] FIG. 17 shows the flow of processing in a case wherein the present invention is applied to the key

scheduler.

[0087] On the key input round of the key scheduler, Kc and $\bar{Kc}$ are randomly selected by a switch SW31 with a probability of almost 1/2 and input to a key scheduler 122. The subsequent processing in the key scheduler is the same as key schedule processing in the general DES. An extended key 123 to be output is Kci when the input key is Kc, and $\bar{Kci}$ when the input key is Kci.

[0088] FIG. 18 shows how an extended key influenced by a mask is applied to a message in each round function.

[0089] A method of applying Kci or $\bar{Kci}$ to a message is generally the same as the method of applying Ki to a message. An exclusive OR 132 applies the extended key Kci or $\bar{Kci}$ to the 48 bits output from an expansion E 131 in units of bits by exclusive OR operation. Since the resultant data is influenced by the mask $\underline{c}$ or $\bar{c}$, if this data is input to an S-box without any change, correct encryption cannot be performed. For this reason, the influence of the mask $\underline{c}$ or $\bar{c}$ on the data must be removed before it is input to the S-box. More specifically, if the influence of the mask is represented by ci, ci is applied to the data by an exclusive OR 133 before the data is input to an S-box 134. Since inversion of an exclusive OR is an exclusive OR, the influence of ci can be removed. This applies to a case wherein the influence of the masks is represented by $\bar{ci}$.

[0090] In this embodiment, if the mask $\bar{c}$ is selected as bit translation of the mask $\underline{c}$, the respective bits of the extended key uniformly take the values "1" and "0". This can prevent leakage of information about the key against various types of observation from outside the encryption apparatus. To minimize leakage of information, ci and $\bar{ci}$ preferably have similar Hamming weights. Note, however, that ci is obtained by processing $\underline{c}$ through a key schedule. It is therefore difficult to completely control the Hamming weights of ci on all the rounds. Under the circumstances, a method of selecting a mask having a Hamming weight corresponding to 1/2 the bit size as the original mask $\underline{c}$ may be used.

[0091] FIG. 19 is a flow chart showing the flow of processing in an encryption method according to an embodiment, which includes the step of masking bits dependent on a plaintext with selected mask patterns and the step of removing the influence of the masks described above from the ciphertext before it is output.

[0092] When plaintext data is input (step U1), at least one i-th mask pair is selected (step U2). With this operation, mask patterns ai (step U3) or inverted mask patterns $\bar{a}$ of the mask patterns ai are selected. The data is masked with the selected masks (step U5). It is checked whether the next mask pair is selected (step U6). If the selection of the next masks are required, the processing is repeated from the step of selecting the new i-th mask pair (step U2). If the selection of the required mask pair is complete, an encryption process of the data is performed (step U7).

[0093] Since the intermediate output data obtained by the encryption process (step U7) has been masked with the mask patterns, the i-th mask pair is determined first (step U8) to determine whether the mask patterns ai were used (step U9) or the inverted mask patterns $\bar{a}$ were used (step U10). The masks applied to the data are removed (step U11). It is then checked whether mask removal is complete (step U12). If masks are left, the processing is repeated from the step of determining the new mask pair (step 8). If mask removal is completed by the above steps, the ciphertext is output (step U13).

[0094] FIG. 20 is a flow chart showing the flow of processing in an encryption method according to an embodiment, which includes the step of removing the influence of masks from input data to a data translation and the step of masking the output data from the data translation with mask patterns.

[0095] When data is input to the data translation (step V1), an i-th mask pair is checked (step V2) to determine whether mask patterns ai were used (step V3) or inverted mask patterns $\bar{a}$ of the mask patterns ai were used (step V4). The masks applied to the data are removed (step V5).

[0096] It is checked whether mask removal is complete (step V6). If masks are left, the processing is repeated from the step of checking a new mask pair (step V2). If mask removal is completed by the above steps, data translation is performed (step V7).

[0097] For the output data upon the above data translation (step V7), at least one i-th mask pair is selected (step V8), and the mask patterns ai (step V9) or mask patterns $\bar{a}$ (step V10) are selected. The data is masked with the selected masks (step V11). It is then checked whether the next mask pair is selected (step V12). If selection of a mask pair that demands selection of the next mask and masking are complete, the data is output from the data translation (step V13).

[0098] FIG. 21 is a flow chart showing the flow of processing in an encryption method according to an embodiment, which includes the step of removing the influence of masks from intermediate bit data during an encryption procedure and the step of masking the data with mask patterns.

[0099] When ciphertext intermediate value as intermediate encryption bit data is input (step W1), an i-th mask pair is checked (step W2) to determine whether mask patterns ai were used (step W3) or inverted mask patterns $\bar{a}$ of the mask patterns ai were used (step W4). The masks applied to the data are removed (step W5).

[0100] It is then checked whether mask removal is complete (step W6). If masks are left, the processing is repeated from the step of checking a new mask pair (step W2). When mask removal is completed by the above steps, an encryption process is performed by an expansion E round function (step W7).

[0101] For the output data from the encryption round function (step 7W), at least one i-th mask pair is selected to select the mask patterns ai (step W9) or the inverted

mask patterns $\bar{a}$ (step W10). The data is masked with the selected mask pair (step W11). It is further checked whether the next mask pair is selected (step W12). If selection of a mask pair that demands selection of the next mask and masking are complete, the ciphertext intermediate value is output (step W13).

[0102] FIG. 22 is a flow chart associated with an encryption procedure according to an embodiment of the present invention. When a plaintext is input (step X1), mask patterns for masking the plaintext is selected (step X2). Bits dependent on the plaintext are masked with the selected mask patterns (step X3).

[0103] For an intermediate value of the encryption data having undergone the above masking process (step X4), mask patterns for masking the input data of a round function is selected (step X5). The masks are removed from the input data of the round function (step X6). Mask patterns for masking an input to the data translation are selected (step X7). The masks are removed from the input data to the data translation (step X8). The data translation then converts the input data (step X9).

[0104] Mask patterns for masking the output from the data translation (step X9) are selected (step X10), and the output data from the data translation is masked with the mask patterns (step X11). Mask patterns for masking the output data of the round function are selected (step X12), and the output data of the round function is masked with the mask patterns (step X13).

[0105] It is checked whether the above procedure is complete up to the nth round (step X14). If the processing is not complete, the processing is repeated from step X4. If the processing is complete up to the nth round, mask patterns that mask the ciphertext are selected (step X15), and the masks are removed from the bits dependent on the ciphertext (step X16). The finally obtained ciphertext is output (step X17).

[0106] As the processing in steps X2, X3, X15, and X16, the processing described with reference to FIG. 19 is performed. As the processing in steps X5, X6, X12, and X13, the processing described with reference to FIG. 20 is performed. As the processing from step S7 to step X11, mask determination processing, mask removal, and concealment processing using masks are performed in one process by using tables calculated in advance and the like to prevent leakage of intermediate data in process.

[0107] FIG. 23 is a block diagram showing the arrangement of an IC card that implements the encryption/decryption apparatus, encryption/decryption method, and program storage medium therefor according to the present invention described above. As shown in FIG. 23, an IC card 201 includes a CPU 203, RAM 205, ROM 207, EEPROM 209, and contactor 211. The RAM 205 is used to store various data and as a work area or the like. The ROM 207 is used to store various data, programs, and the like. The EEPROM 209 is used to store the programs indicated by the flow charts of FIGS. 19 to 22 and the like. The contactor 211 obtains electrical contact with an IC card reader/writer (not shown). Note that the programs shown in FIGS. 19 to 22 may be stored in the RAM 205 or ROM 207 instead of the EEPROM 209.

[0108] In the above embodiment, the application of the present invention to the DES scheme has been described in detail. However, the present invention is not limited to this and can be applied to general encryption schemes comprised of part or all of the following three types of functions, namely dyadic operation like exclusive OR operation, a permutation equivalent to bit interchange, and cipher system equivalent an S-box.

Claims

1. An encryption apparatus for converting a plaintext block into a ciphertext block depending on supplied key information, characterized by comprising:

   means (SW1, SW2) for randomly selecting one pattern of each of pairs $a_i,\bar{a}_i$ (where i is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time encryption is performed;
   means (43a, 44a, 42a) for masking bits dependent on a plaintext within said apparatus with the mask patterns selected by said selection means; and
   means (42b, 43b, 44b) for removing an influence of the mask $\underline{a}$ from a ciphertext before the ciphertext is output.

2. An encryption apparatus for converting a plaintext block into a ciphertext block depending on supplied key information, characterized by comprising:

   means (SW1, SW2) for randomly selecting one pattern of each of pairs $a_i,\bar{a}_i$ (where i is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time encryption is performed;
   means (33a, 33b) for masking intermediate bit data within said apparatus with the mask patterns selected by said selection means; and
   means (32a, 33b) for removing an influence of the mask $\underline{a}$ from the intermediate bit data masked by said masking means.

3. An encryption apparatus for converting a plaintext block into a ciphertext block depending on supplied key information, characterized by comprising:

   data translation means (17, 35) for performing

data translation to intermediate data within said apparatus;
means (SW13, SW23) for randomly selecting one pattern of each of pairs ai, $\bar{ai}$ (where i is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time encryption is performed;
means (33a, 33b) for masking an input to said data translation means with the mask patterns selected by said selection means; and
means (24, 25) for removing an influence of the mask a from an output from said data translation means which is masked by said masking means.

4. An apparatus according to claim 1, characterized in that said means (43a, 44a, 42a) for masking the bits dependent on the plaintext within said apparatus with the selected mask patterns and said means (42b, 43b, 44b) for removing the influence of the mask a from the ciphertext comprise one of an exclusive OR, addition or subtraction with respect to a modulus w, and multiplication or division with respect to the modulus w.

5. An apparatus according to claim 2, characterized in that said means (33a, 33b) for masking the intermediate bit data within said apparatus with the selected mask patterns and said means (32a, 33b) for removing the influence of the mask a from the masked intermediate bit data comprise one of an exclusive OR, addition or subtraction with respect to a modulus w, and multiplication or division with respect to the modulus w.

6. An apparatus according to claim 3, characterized in that said data translation means (17, 35), said means (33a, 33b) for masking the input to said data translation means (17, 35) with the selected mask patterns, and said means (24, 25) for removing the influence of the mask a from the masked output from said data translation means (17, 35) comprise one of an exclusive OR, addition or subtraction with respect to a modulus w, and multiplication or division with respect to the modulus w.

7. An apparatus according to claim 3, characterized by further comprising:

first storage means (34a) for storing, in the form of a table, said means (SW13, SW23) for randomly selecting one pattern of each of the pairs ai, $\bar{ai}$ (where i is a positive integer not less than one) of one or the plurality of predetermined mask patterns and the mask patterns obtained by bit inversion of the predetermined mask patterns every time encryption is performed, said means (33a) for masking the input to said data translation means with the mask patterns ai, and said means (24, 25) for removing the influence of the masks ai from the masked output from said data translation means;
second storage means (34b) for storing, in the form of a table, said means (33b) for masking the input to said data translation means with mask patterns $\bar{a}$, and said means (24, 25) for removing an influence of the masks $\bar{a}$ from the masked output from said data translation means; and
masked data translation means (EEPROM, RAM, ROM) for randomly selecting one of said first and second storage means every time encryption is performed, and performing the processing by said data translation means for masked data.

8. An apparatus according to claim 1, characterized in that the pair a,$\bar{a}$ of the mask patterns and the mask patterns obtained by bit inversion comprises a pair a,$\bar{a}$ of predetermined fixed mask patterns and mask patterns obtained by bit inversion of the fixed mask patterns.

9. An apparatus according to claim 1, characterized in that the pair a,$\bar{a}$ of the mask patterns and the mask patterns obtained by bit inversion are not necessarily concealed.

10. An apparatus according to claim 1, characterized in that a Hamming weight indicating the number of bits "1s" of an n-bit long bit sequence x is defined as H (x), and the Hamming weight H(a) of the mask a satisfies $0 < H(a) < n$.

11. An apparatus according to claim 1, characterized in that a Hamming weight indicating the number of bits "1s" of an n-bit long bit sequence x is defined as H (x), and an absolute value of a difference between the Hamming weight H(a) of the mask a and a Hamming weight H($\bar{a}$) of bit inversion $\bar{a}$ of the mask a is less than n/2.

12. A decryption apparatus for converting a ciphertext block into a plaintext block depending on supplied key information, characterized by comprising:

means (SW21, SW22) for randomly selecting one pattern of each of pairs ai,$\bar{ai}$ (where i is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time decryption is performed;
means (43a, 44a, 42a) for masking bits de-

pendent on a ciphertext within said apparatus with the mask patterns selected by said selection means; and
means (42b, 43b, 44b) for removing an influence of the mask $\underline{a}$ from a plaintext before the plaintext is output.

13. A decryption apparatus for converting a ciphertext block into a plaintext block depending on supplied key information, characterized by comprising:

means (SW1, SW2) for randomly selecting one pattern of each of pairs $ai,\bar{ai}$ (where $\underline{i}$ is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time decryption is performed;
means (33a, 33b) for masking intermediate bit data within said apparatus with the mask patterns selected by said selection means; and
means (32a, 33b) for removing an influence of the mask $\underline{a}$ from the intermediate bit data masked by said masking means.

14. A decryption apparatus for converting a ciphertext block into a plaintext block depending on supplied key information, characterized by comprising:

data translation means (17, 35) for performing data translation to intermediate data within said apparatus;
means (SW13, SW23) for randomly selecting one pattern of each of pairs ai, $\bar{ai}$ (where $\underline{i}$ is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time decryption is performed;
means (33a, 33b) for masking an input to said data translation means with the mask patterns selected by said selection means; and
means (24, 25) for removing an influence of the mask $\underline{a}$ from an output from said data translation means which is masked by said masking means.

15. An apparatus according to claim 12, characterized in that said means (43a, 44a, 42a) for masking the bits dependent on the plaintext within said apparatus with the selected mask patterns and said means (24, 25) for removing the influence of the mask $\underline{a}$ from the ciphertext comprise one of an exclusive OR, addition or subtraction with respect to a modulus $\underline{w}$, and multiplication or division with respect to the modulus $\underline{w}$.

16. An apparatus according to claim 13, characterized in that said means (33a, 33b) for masking the intermediate bit data within said apparatus with the selected mask patterns and said means (32a, 33b) for removing the influence of the mask $\underline{a}$ from the masked intermediate bit data comprise one of an exclusive OR, addition or subtraction with respect to a modulus $\underline{w}$, and multiplication or division with respect to the modulus $\underline{w}$.

17. An apparatus according to claim 15, characterized in that said data translation means (17, 35), said means (33a, 33b) for masking the input to said data translation means (17, 35) with the selected mask patterns, and said means (24, 25) for removing the influence of the mask $\underline{a}$ from the masked output from said data translation means comprise one of an exclusive OR, addition or subtraction with respect to a modulus $\underline{w}$, and multiplication or division with respect to the modulus $\underline{w}$.

18. An apparatus according to claim 14, characterized by further comprising:

first storage means (34a) for storing, in the form of a table, said means (SW13, SW23) for randomly selecting one pattern of each of the pairs $ai,\bar{ai}$ (where $\underline{i}$ is a positive integer not less than one) of one or the plurality of predetermined mask patterns and the mask patterns obtained by bit inversion of the predetermined mask patterns every time decryption is performed, said means (33a, 33b) for masking the input to said data translation means with the mask patterns ai, and means (24, 25) for removing the influence of the masks ai from the masked output from said data translation means;
second storage means (34b) for storing, in the form of a table, means (33b) for masking the input to said data translation means with mask patterns $\bar{a}$, and means (24, 25) for removing an influence of the masks $\bar{a}$ from the masked output from said data translation means; and
masked data translation means (EEPROM, RAM, ROM) for randomly selecting one of said first and second storage means every time decryption is performed, and performing the processing by said data translation means for masked data.

19. An apparatus according to claim 12, characterized in that the pair $a,\bar{a}$ of the mask patterns and the mask patterns obtained by bit inversion comprises a pair $a,\bar{a}$ of predetermined fixed mask patterns and mask patterns obtained by bit inversion of the fixed mask patterns.

20. An apparatus according to claim 13, characterized in that the pair $ai,\bar{ai}$ of the mask patterns and the

mask patterns obtained by bit inversion are not necessarily concealed.

21. An apparatus according to claim 12, characterized in that a Hamming weight indicating the number of bits "1s" of an n-bit long bit sequence $\underline{x}$ is defined as H(x), and the Hamming weight H(a) of the mask $\underline{a}$ satisfies $0 < H(a) < n$.

22. An apparatus according to claim 12, characterized in that a Hamming weight indicating the number of bits "1s" of an n-bit long bit sequence $\underline{x}$ is defined as H(x), and an absolute value of a difference between the Hamming weight H(a) of the mask $\underline{a}$ and a Hamming weight $H(\bar{a})$ of bit inversion $\bar{a}$ of the mask $\underline{a}$ is less than n/2.

23. An encryption method of converting a plaintext block into a ciphertext block depending on supplied key information, characterized by comprising the steps of:

(U2, U3, U4) randomly selecting one pattern of each of pairs $ai,\bar{ai}$ (where $\underline{i}$ is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time encryption is performed;
(U5) masking bits dependent on a plaintext within the method with the selected mask patterns; and
(U11) removing an influence of the mask $\underline{a}$ from a ciphertext before the ciphertext is output.

24. An encryption method of converting a plaintext block into a ciphertext block depending on supplied key information, characterized by comprising the steps of:

(W2, W3, W4) randomly selecting one pattern of each of pairs $ai,\bar{ai}$ (where $\underline{i}$ is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time encryption is performed;
(W5) masking intermediate bit data within the method with the selected mask patterns; and
(W11) removing an influence of the mask $\underline{a}$ from the masked intermediate bit data.

25. An encryption method of converting a plaintext block into a ciphertext block depending on supplied key information, characterized by comprising the steps of:

(V7) performing data translation to intermediate data within the method;
(V8, V9, V10) randomly selecting one pattern of each of pairs $ai,\bar{ai}$ (where $\underline{i}$ is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time encryption is performed;
(V11) masking an input to the data translation step with the selected mask patterns; and
(V5) removing an influence of the mask $\underline{a}$ from a masked output from the data translation step.

26. A method according to claim 23, characterized in that the step of masking the bits dependent on the plaintext within the method with the selected mask patterns and the step of removing the influence of the mask $\underline{a}$ from the ciphertext comprise one of an exclusive OR, addition or subtraction with respect to a modulus $\underline{w}$, and multiplication or division with respect to the modulus $\underline{w}$.

27. A method according to claim 24, characterized in that the step of masking the intermediate bit data within the method with the selected mask patterns and the step of removing the influence of the mask $\underline{a}$ from the masked intermediate bit data comprise one of an exclusive OR, addition or subtraction with respect to a modulus $\underline{w}$, and multiplication or division with respect to the modulus $\underline{w}$.

28. A method according to claim 25, characterized in that the data translation step, the step of masking the input to the data translation step with the selected mask patterns, and the step of removing the influence of the mask $\underline{a}$ from the masked output from the data translation step comprise one of an exclusive OR, addition or subtraction with respect to a modulus $\underline{w}$, and multiplication or division with respect to the modulus $\underline{w}$.

29. A method according to claim 25, characterized by further comprising the steps of:

storing, in the form of a table, the step of randomly selecting one pattern of each of the pairs $ai,\bar{ai}$ (where $\underline{i}$ is a positive integer not less than one) of one or the plurality of predetermined mask patterns and the mask patterns obtained by bit inversion of the predetermined mask patterns every time encryption is performed, the step of masking the input to said data translation step with the mask patterns ai, and the step of removing the influence of the masks ai from the masked output from the data translation step;
storing, in the form of a table, the step of masking the input to said data translation step with

mask patterns $\bar{a}$, and step of removing an influence of the masks $\bar{a}$ from the masked output from the data translation step; and
randomly selecting one of the first and second storage steps every time encryption is performed, and performing the processing in the data translation step for masked data.

30. A method according to claim 23, characterized in that the pair a,$\bar{a}$ of the mask patterns and the mask patterns obtained by bit inversion comprises a pair a,$\bar{a}$ of predetermined fixed mask patterns and mask patterns obtained by bit inversion of the fixed mask patterns.

31. A method according to claim 23, characterized in that the pair a,$\bar{a}$ of the mask patterns and the mask patterns obtained by bit inversion are not necessarily concealed.

32. A method according to claim 23, characterized in that a Hamming weight indicating the number of bits "1s" of an n-bit long bit sequence $\underline{x}$ is defined as H (x), and the Hamming weight H(a) of the mask $\underline{a}$ satisfies $0 < H(a) < n$.

33. A method according to claim 23, characterized in that a Hamming weight indicating the number of bits "1s" of an n-bit long bit sequence $\underline{x}$ is defined as H (x), and an absolute value of a difference between the Hamming weight H(a) of the mask $\underline{a}$ and a Hamming weight H($\bar{a}$) of bit inversion $\bar{a}$ of the mask $\underline{a}$ is less than n/2.

34. A decryption method of converting a ciphertext block into a plaintext block depending on supplied key information, characterized by comprising the steps of:

(U2, U3, U4) randomly selecting one pattern of each of pairs ai,$\bar{a}$i (where $\underline{i}$ is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time decryption is performed;
(U5) masking bits dependent on a ciphertext within the method with the selected mask patterns; and
(U11) removing an influence of the mask $\underline{a}$ from a plaintext before the plaintext is output.

35. A decryption method of converting a ciphertext block into a plaintext block depending on supplied key information, characterized by comprising the steps of:

(W2, W3, W4) randomly selecting one pattern of each of pairs ai, $\bar{a}$i (where $\underline{i}$ is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time decryption is performed;
(W5) masking intermediate bit data within the method with the selected mask patterns; and
(W11) removing an influence of the mask $\underline{a}$ from the masked intermediate bit data.

36. A decryption method of converting a ciphertext block into a plaintext block depending on supplied key information, characterized by comprising the steps of:

(V7) performing data translation to intermediate data within the method;
(V8, V9, V10) randomly selecting one pattern of each of pairs ai,$\bar{a}$i (where $\underline{i}$ is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time decryption is performed;
(V11) masking an input to the data translation step with the selected mask patterns; and
(V5) removing an influence of the mask $\underline{a}$ from a masked output from the data translation step.

37. A method according to claim 34, characterized in that the step of masking the bits dependent on the ciphertext within the method with the selected mask patterns and the step of removing the influence of the mask $\underline{a}$ from the plaintext comprise one of an exclusive OR, addition or subtraction with respect to a modulus $\underline{w}$, and multiplication or division with respect to the modulus $\underline{w}$.

38. A method according to claim 35, characterized in that the step of masking the intermediate bit data within the method with the selected mask patterns and the step of removing the influence of the mask $\underline{a}$ from the masked intermediate bit data comprise one of an exclusive OR, addition or subtraction with respect to a modulus $\underline{w}$, and multiplication or division with respect to the modulus $\underline{w}$.

39. A method according to claim 36, characterized in that the data translation step, the step of masking the input to the data translation step with the selected mask patterns, and the step of removing the influence of the mask $\underline{a}$ from the masked output from the data translation step comprise one of an exclusive OR, addition or subtraction with respect to a modulus $\underline{w}$, and multiplication or division with respect to the modulus $\underline{w}$.

40. A method according to claim 36, characterized by further comprising the steps of:

storing, in the form of a table, the step of randomly selecting one pattern of each of the pairs ai,$\bar{ai}$ (where i is a positive integer not less than one) of one or the plurality of predetermined mask patterns and the mask patterns obtained by bit inversion of the predetermined mask patterns every time decryption is performed, the step of masking the input to said data translation step with the mask patterns ai, and the step of removing the influence of the masks ai from the masked output from the data translation step;
storing, in the form of a table, the step of masking the input to said data translation step with mask patterns $\bar{a}$, and step of removing an influence of the masks $\bar{a}$ from the masked output from the data translation step; and
randomly selecting one of the first and second storage steps every time decryption is performed, and performing the processing in the data translation step for masked data.

41. A method according to claim 34, characterized in that the pair a,$\bar{a}$ of the mask patterns and the mask patterns obtained by bit inversion comprises a pair a,$\bar{a}$ of predetermined fixed mask patterns and mask patterns obtained by bit inversion of the fixed mask patterns.

42. A method according to claim 34, characterized in that the pair a,$\bar{a}$ of the mask patterns and the mask patterns obtained by bit inversion are not necessarily concealed.

43. A method according to claim 34, characterized in that a Hamming weight indicating the number of bits "1s" of an n-bit long bit sequence x is defined as H (x), and the Hamming weight H(a) of the mask a satisfies $0 < H(a) < n$.

44. A method according to claim 34, characterized in that a Hamming weight indicating the number of bits "1s" of an n-bit long bit sequence x is defined as H (x), and an absolute value of a difference between the Hamming weight H(a) of the mask a and a Hamming weight H($\bar{a}$) of bit inversion $\bar{a}$ of the mask a is less than n/2.

45. A computer-usable program storage medium (205, 207, 209) storing computer-readable program code means for converting a plaintext block into a ciphertext block depending on supplied key information, characterized by comprising:

computer-readable program code means (U2, U3, U4) for causing a computer to randomly select one pattern of each of pairs ai,$\bar{ai}$ (where i is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time encryption is performed;
computer-readable program code means (U5) for causing said computer to mask bits dependent on a plaintext within the method with the selected mask patterns; and
computer-readable program code means (U11) for causing said computer to remove an influence of the mask a from a ciphertext before the ciphertext is output.

46. An encryption apparatus for converting a plaintext block into a ciphertext block depending on supplied key information, characterized by comprising:

means (SW31) for randomly selecting one pattern of each of pairs ai,$\bar{ai}$ (where i is a positive integer not less than one) of one or a plurality of predetermined mask patterns and mask patterns obtained by bit inversion of the predetermined mask patterns every time encryption is performed;
means (122) for masking bits dependent on a key within said apparatus with the mask patterns selected by said selection means;
data translation means (132) for converting intermediate data within said apparatus with the key; and
means (133) for removing an influence of the mask a from an output from said data translation means.

47. An apparatus according to claim 46, characterized in that the pair a,$\bar{a}$ of the mask patterns and the mask patterns obtained by bit inversion comprises a pair a,$\bar{a}$ of predetermined fixed mask patterns and mask patterns obtained by bit inversion of the fixed mask patterns.

48. An apparatus according to claim 46, characterized in that the pair a,$\bar{a}$ of the mask patterns and the mask patterns obtained by bit inversion are not necessarily concealed.

49. An apparatus according to claim 46, characterized in that a Hamming weight indicating the number of bits "1s" of an n-bit long bit sequence x is defined as H(x), and the Hamming weight H(a) of the mask a satisfies $0 < H(a) < n$.

50. An apparatus according to claim 46, characterized in that a Hamming weight indicating the number of bits "1s" of an n-bit long bit sequence x is defined

as H(x), and an absolute value of a difference between the Hamming weight H(a) of the mask $\underline{a}$ and a Hamming weight $H(\bar{a})$ of bit inversion $\bar{a}$ of the mask $\underline{a}$ is less than n/2.

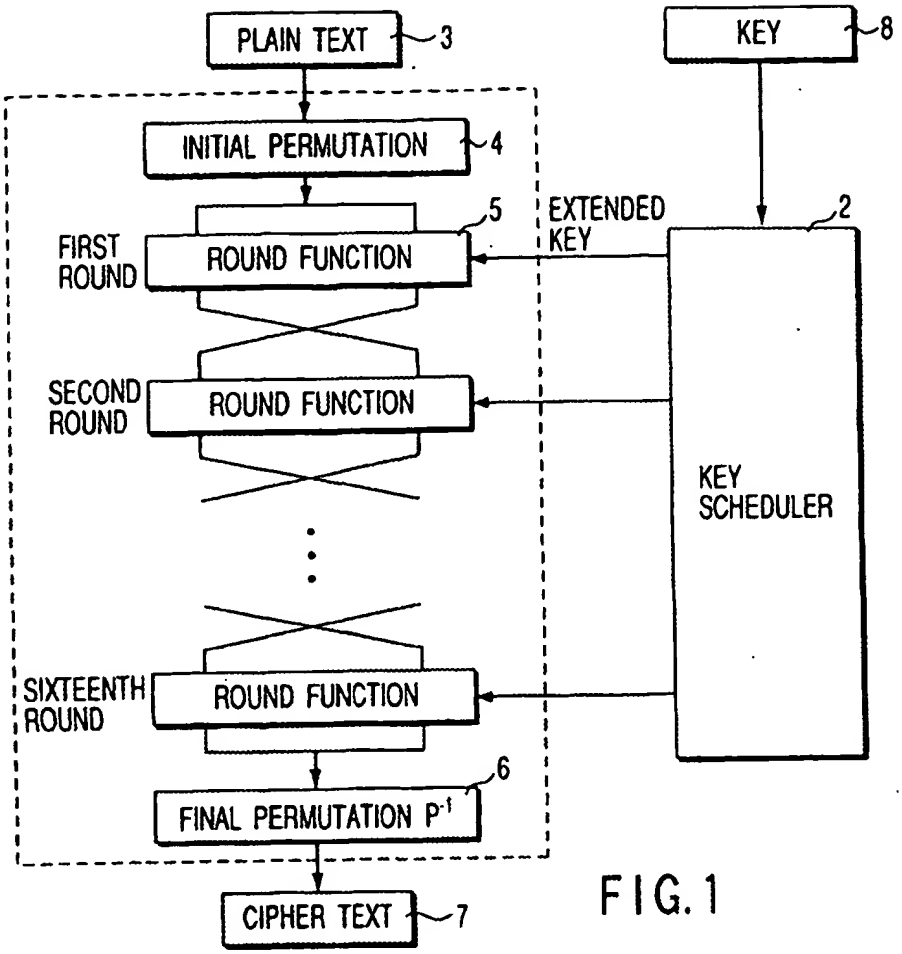

PLAIN TEXT
3
KEY
8
INITIAL PERMUTATION
4
FIRST ROUND
ROUND FUNCTION
5
EXTENDED KEY
2
SECOND ROUND
ROUND FUNCTION
KEY SCHEDULER
SIXTEENTH ROUND
ROUND FUNCTION
FINAL PERMUTATION P-1
6
CIPHER TEXT
7

FIG. 1

FIG. 2

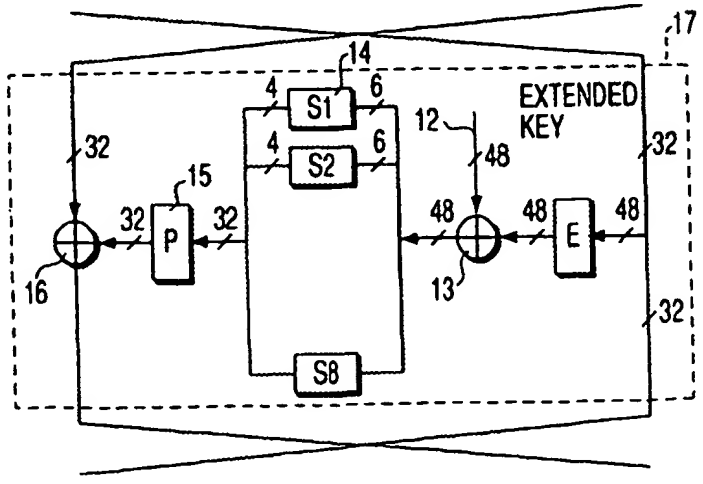

17
14
4
S1
6
EXTENDED KEY
12
48
32
4
S2
6
15
32
P
32
48
48
E
48
32
16
13
32
S8

TABLE OF S1

14,	4,	13,	1,	2,	15,	11,	8,	3,	10,	6,	12,	5,	9,	0,	7,
0,	15,	7,	4,	14,	2,	13,	1,	10,	6,	12,	11,	9,	5,	3,	8,
4,	1,	14,	8,	13,	6,	2,	11,	15,	12,	9,	7,	3,	10,	5,	0,
15,	12,	8,	2,	4,	9,	1,	7,	5,	11,	3,	14,	10,	0,	6,	13.

FIG. 3

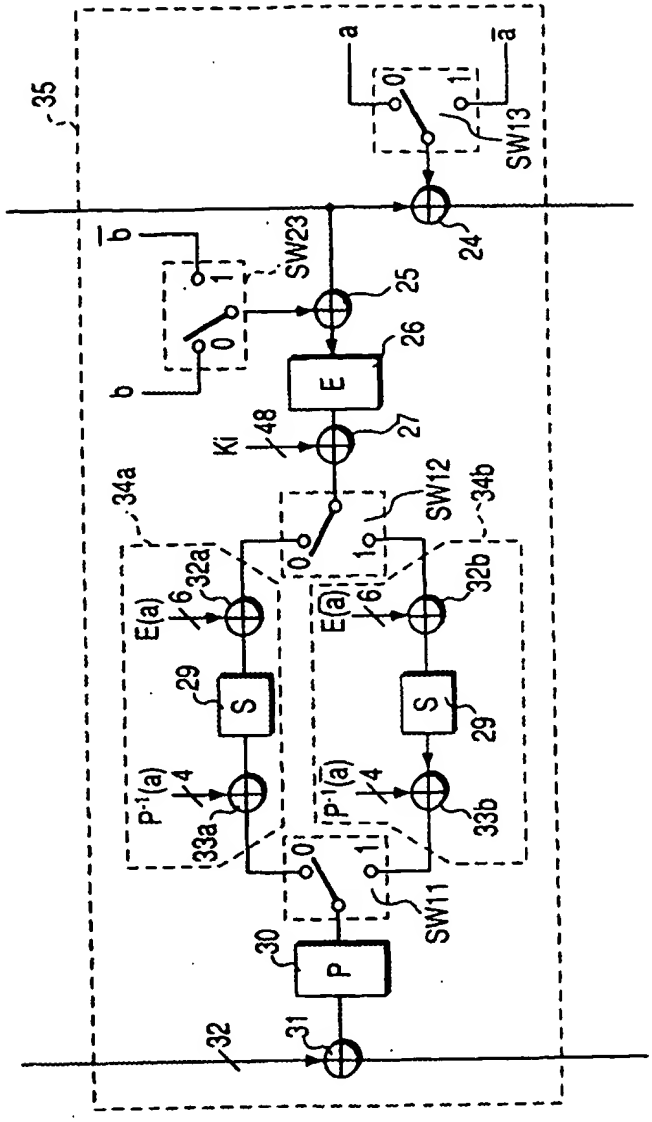


FIG. 4

FIG. 5A

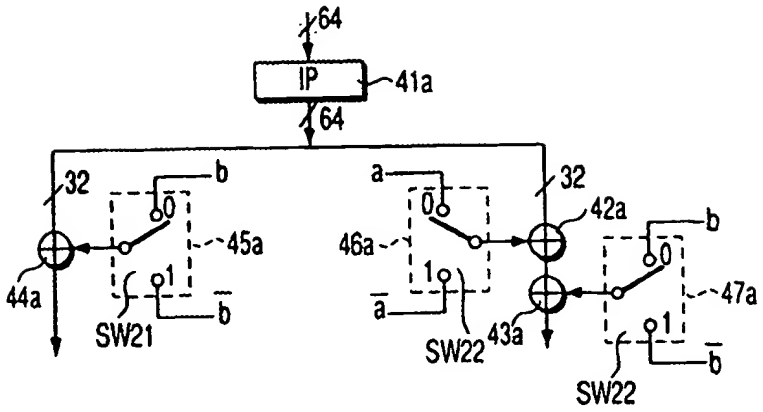


FIG. 5B

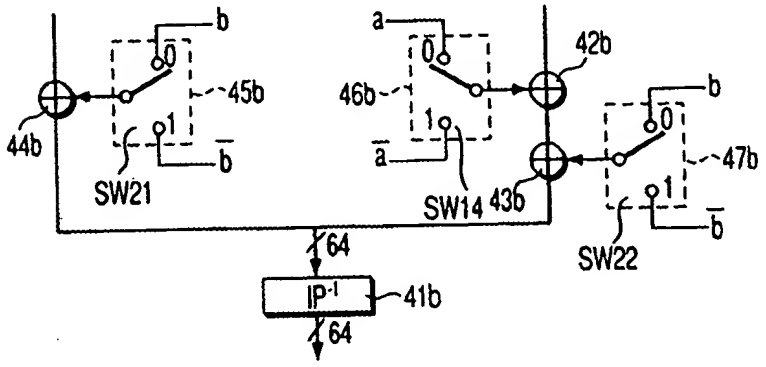


FIG. 6

TABLE OF EXPANSION E

32,	1,	2,	3,	4,	5,
4,	5,	6,	7,	8,	9,
8,	9,	10,	11,	12,	13,
12,	13.	14,	15,	16,	17,
16,	17,	18,	19,	20,	21,
20,	21,	22,	23,	24,	25,
24,	25,	26,	27,	28,	29,
28,	29.	30,	31,	32,	1,

FIG. 7

TABLE OF PERMUTATION P

16,	7.	20,	21,	29.	12,	28,	17,	1,	15,	23,	26,	5,	18,	31,	10,
2,	8,	24,	14,	32.	27,	3,	9,	19,	13,	30,	6,	22,	11,	4,	25,

OUTPUT OF CONCEALED S1 WHEN INPUT CORRESPONDS TO (000000, 000001,···,111111) IN CASE OF MASK a

8	11	14	2	13	4	0	15	1	7	11	1	6	13	5	8	15	12	4	9	3	10	9		
3	10	0	7	14	12	6	2	5	4	2	8	1	7	11	11	16	14	4	13	8	9	7	6	
13	10	9	1	14	0	12	12	0	5	15	2	5	3	10	15	3								

FIG. 8

TABLE OF MASK $\bar{a}$ (BIT INVERSION OF a)

12	0	5	12	10	13	0	10	15	3	3	15	1	14	6	5	2	9	8	6	7	2	11	1	9
4	4	8	14	7	13	11	10	1	3	9	3	1	8	15	5	12	6	5	12	6	11	3	0	7
10	2	9	14	4	8	14	0	15	11	2	13	1	4	7										

FIG. 9

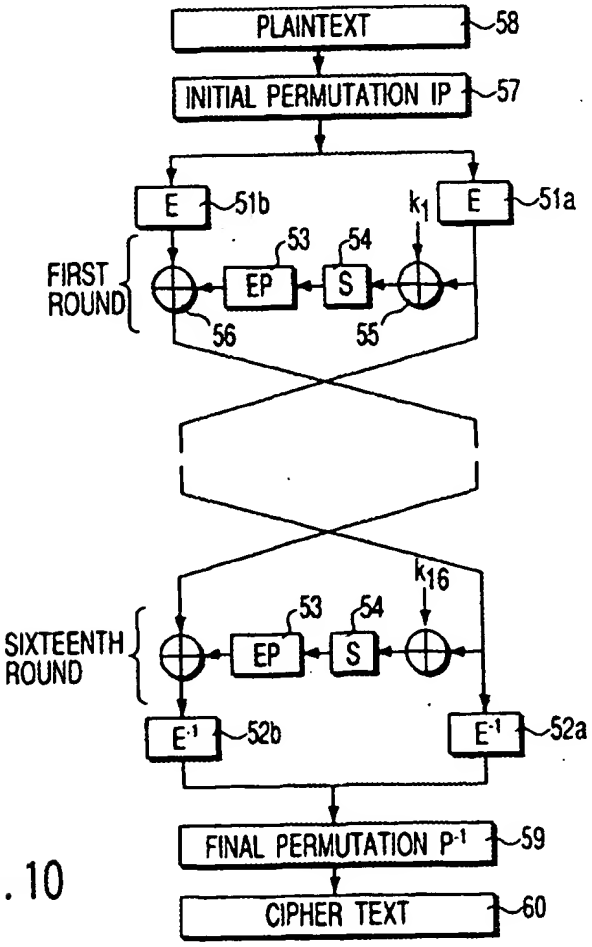


FIG. 10

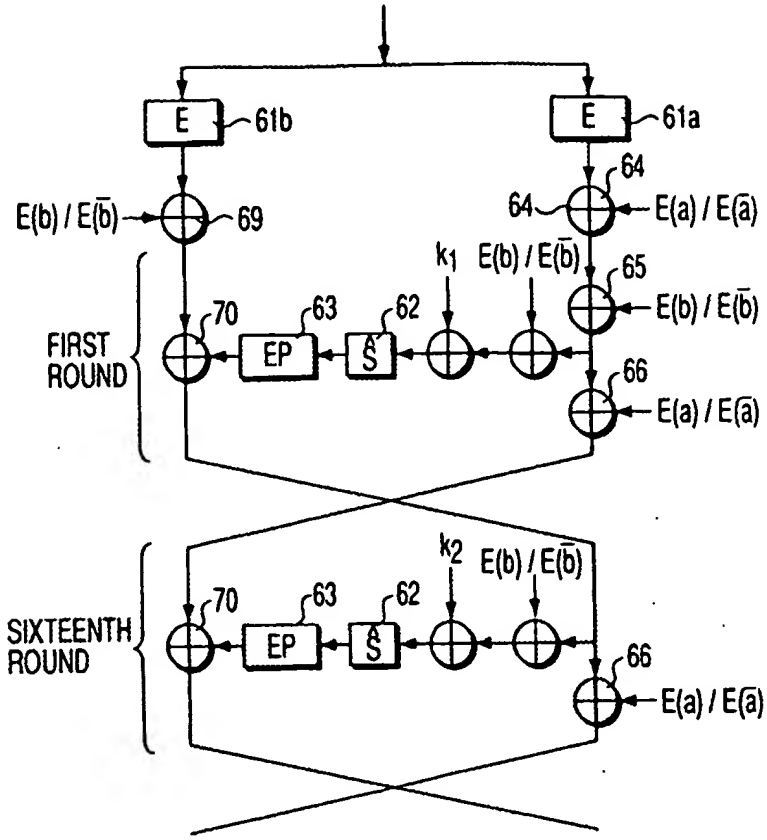

E
61b
E
61a
64
E(b) / E(b̄)
69
64
E(a) / E(ā)
k1
E(b) / E(b̄)
65
E(b) / E(b̄)
70
63
62
FIRST ROUND
EP
S
66
E(a) / E(ā)
k2
E(b) / E(b̄)
70
63
62
SIXTEENTH ROUND
EP
S
66
E(a) / E(ā)

FIG. 11

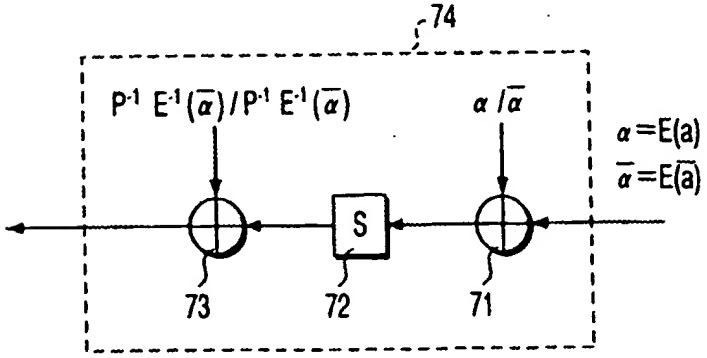

74
P-1 E-1(ᾱ) / P-1 E-1(ᾱ)
α / ᾱ
α=E(a)
ᾱ=E(ā)
S
73
72
71

FIG. 12

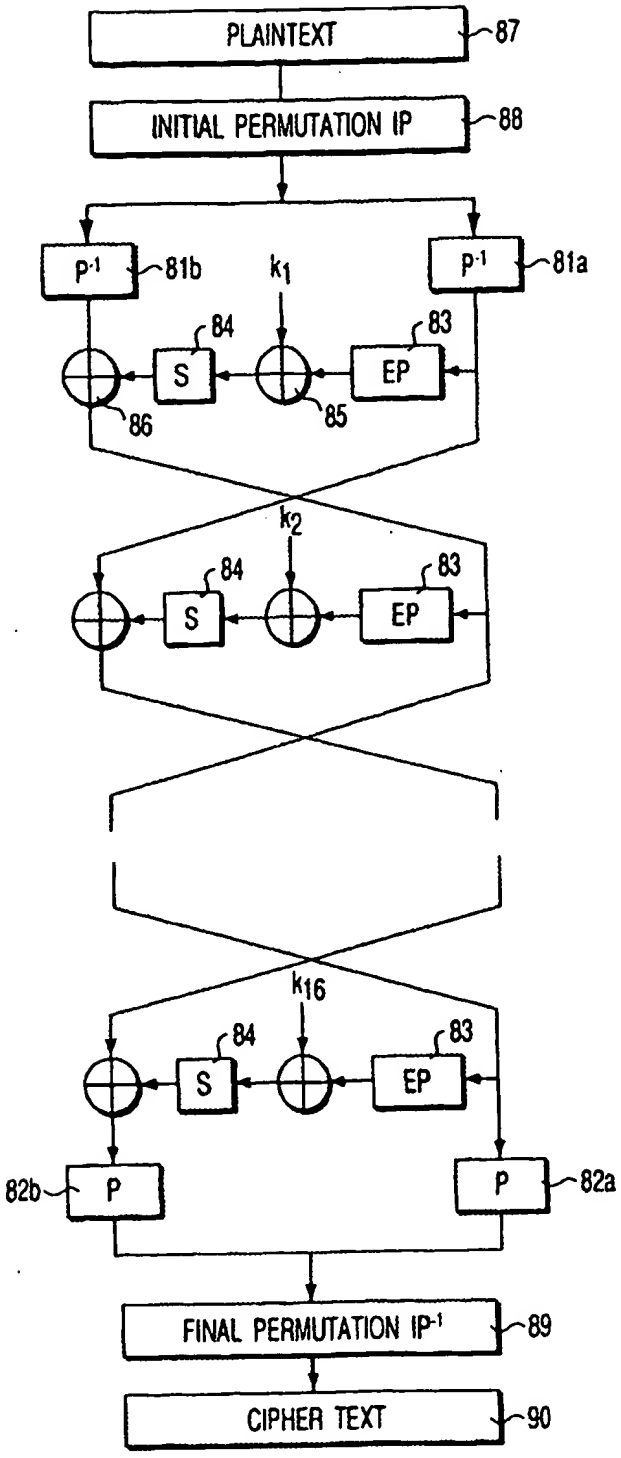

PLAINTEXT
87
INITIAL PERMUTATION IP
88
P-1
81b
k1
P-1
81a
84
S
83
EP
86
85
k2
84
S
83
EP
k16
84
S
83
EP
82b
P
P
82a
FINAL PERMUTATION IP-1
89
CIPHER TEXT
90

FIG. 13

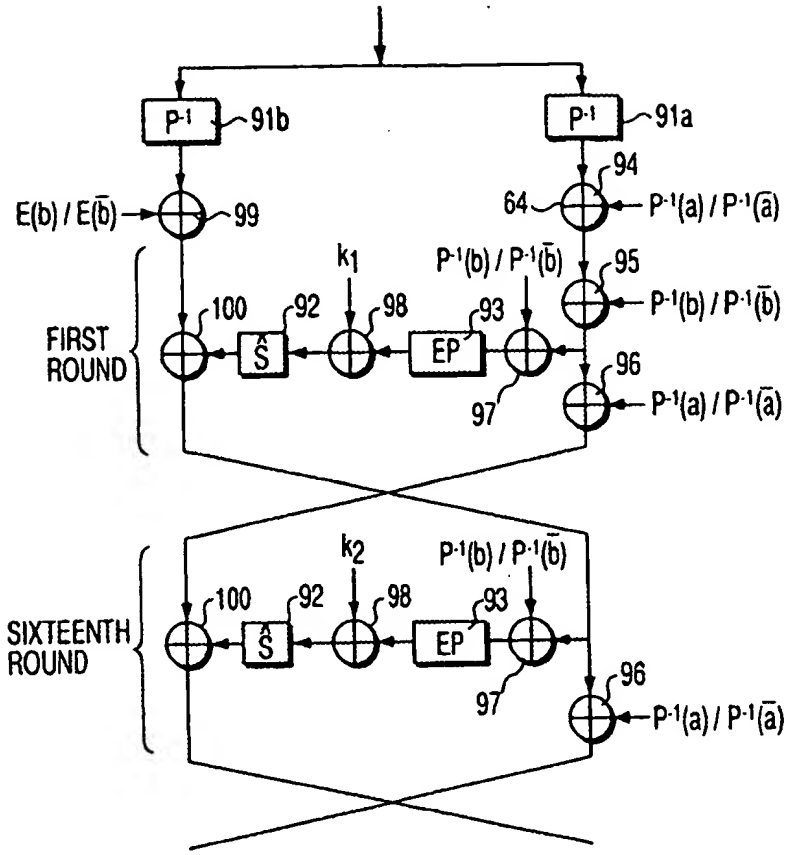

P-1
91b
P-1
91a
94
E(b) / E(b̄)
99
64
P-1(a) / P-1(ā)
k1
P-1(b) / P-1(b̄)
95
P-1(b) / P-1(b̄)
FIRST ROUND
100
92
98
93
EP
96
97
P-1(a) / P-1(ā)
k2
P-1(b) / P-1(b̄)
SIXTEENTH ROUND
100
92
98
93
EP
96
97
P-1(a) / P-1(ā)

FIG. 14

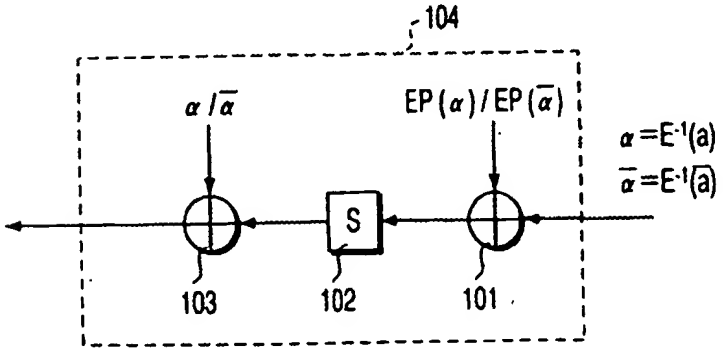

104
α / ᾱ
EP(α) / EP(ᾱ)
α=E-1(a)
ᾱ=E-1(ā)
S
103
102
101

FIG. 15

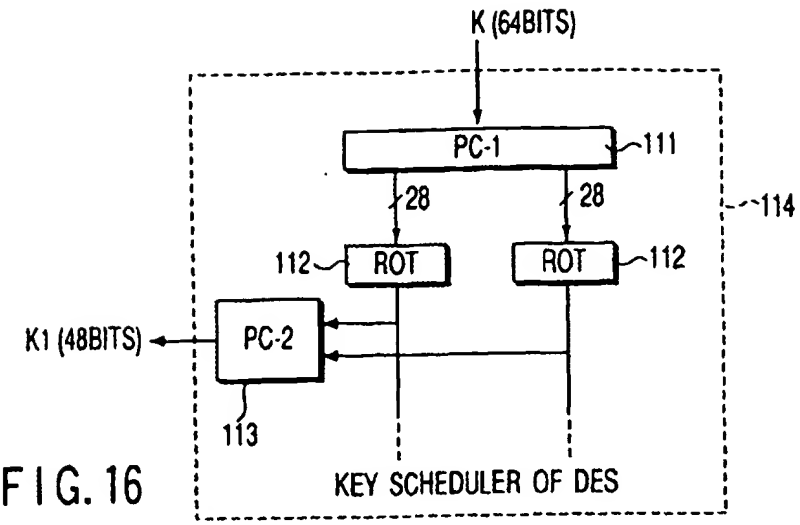


FIG. 16

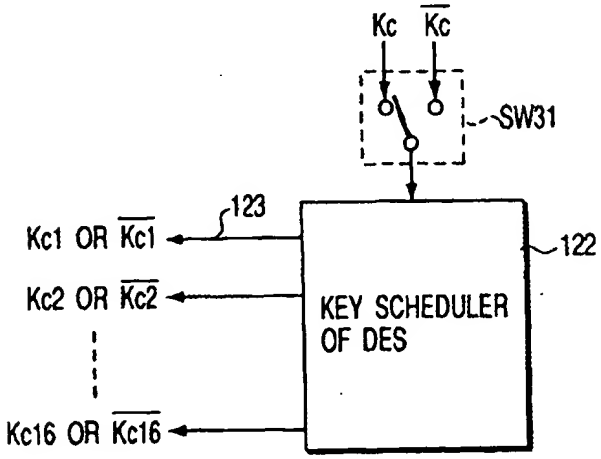


FIG. 17

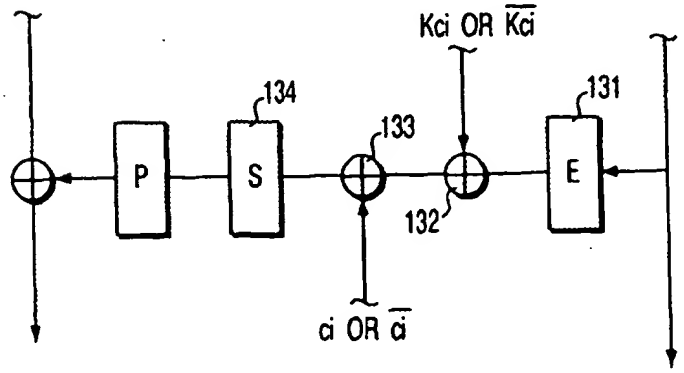


FIG. 18

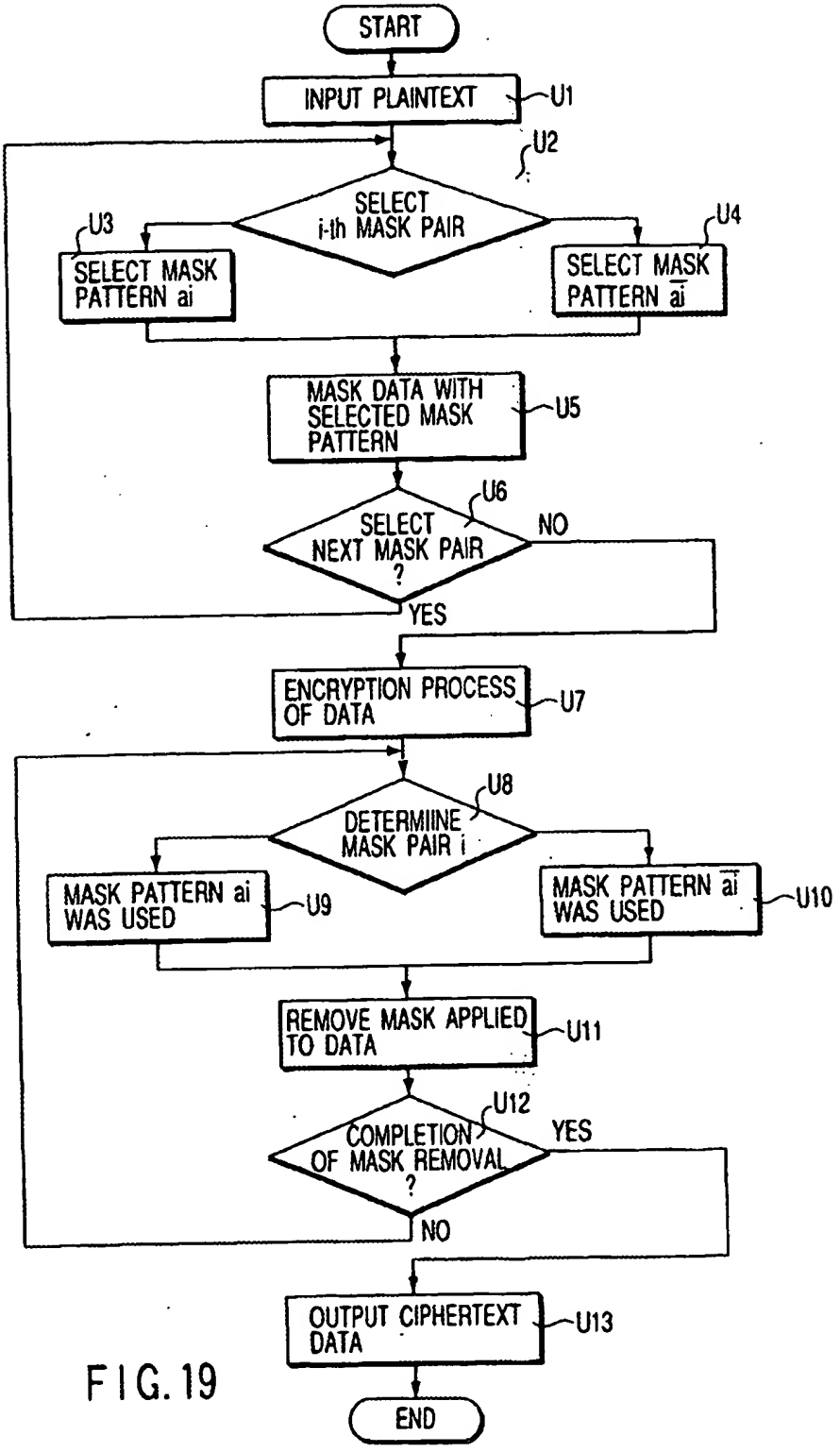

START
INPUT PLAINTEXT
U1
U2
SELECT i-th MASK PAIR
U3
SELECT MASK PATTERN ai
U4
SELECT MASK PATTERN $\overline{ai}$
MASK DATA WITH SELECTED MASK PATTERN
U5
U6
SELECT NEXT MASK PAIR ?
NO
YES
ENCRYPTION PROCESS OF DATA
U7
U8
DETERMIINE MASK PAIR i
MASK PATTERN ai WAS USED
U9
MASK PATTERN $\overline{ai}$ WAS USED
U10
REMOVE MASK APPLIED TO DATA
U11
U12
COMPLETION OF MASK REMOVAL ?
YES
NO
OUTPUT CIPHERTEXT DATA
U13
END

FIG. 19

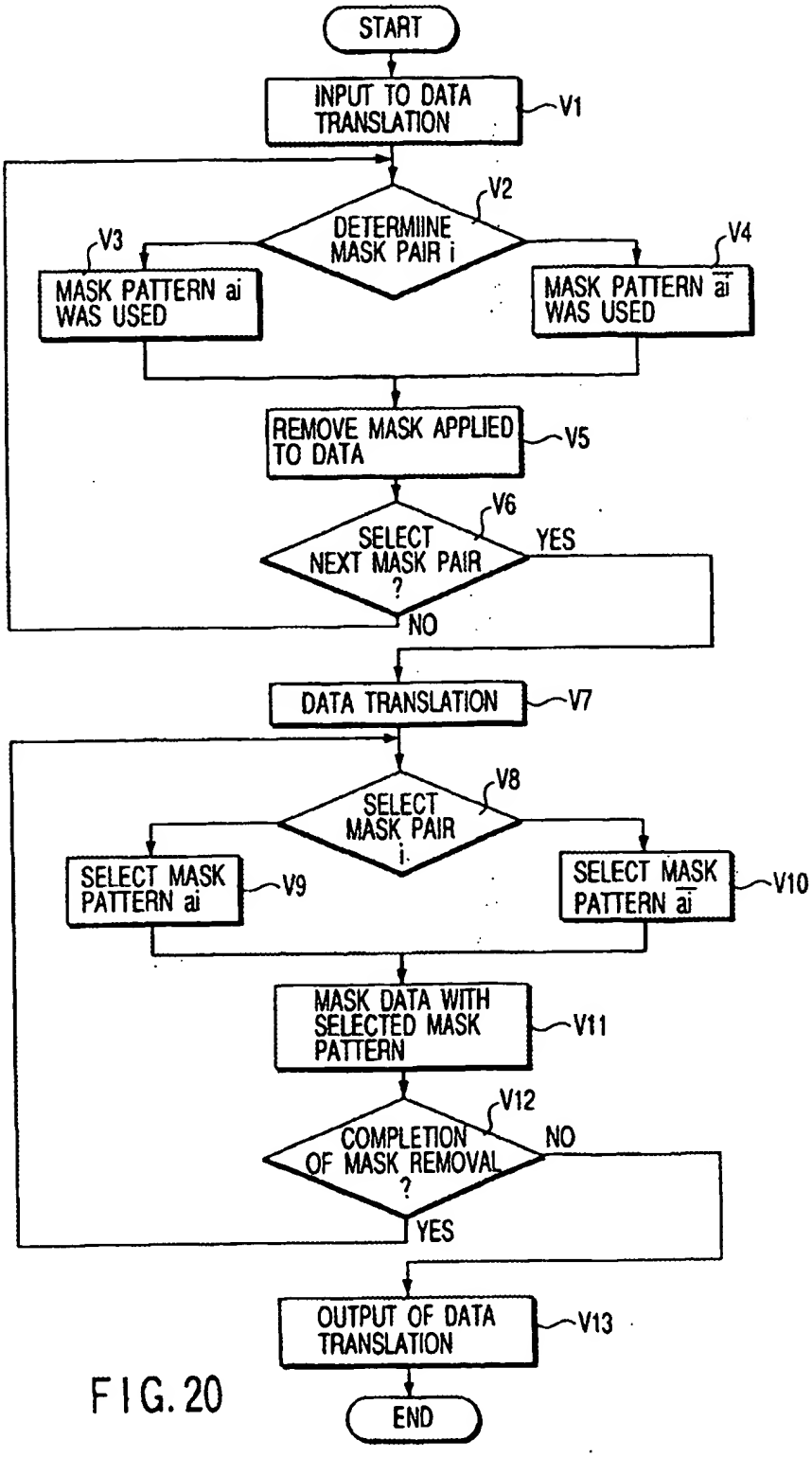

START
INPUT TO DATA TRANSLATION
V1
V2
DETERMIINE MASK PAIR i
V3
MASK PATTERN ai WAS USED
V4
MASK PATTERN $\overline{ai}$ WAS USED
REMOVE MASK APPLIED TO DATA
V5
V6
SELECT NEXT MASK PAIR ?
YES
NO
DATA TRANSLATION
V7
V8
SELECT MASK PAIR i
SELECT MASK PATTERN ai
V9
SELECT MASK PATTERN $\overline{ai}$
V10
MASK DATA WITH SELECTED MASK PATTERN
V11
V12
COMPLETION OF MASK REMOVAL ?
NO
YES
OUTPUT OF DATA TRANSLATION
V13
END

FIG. 20

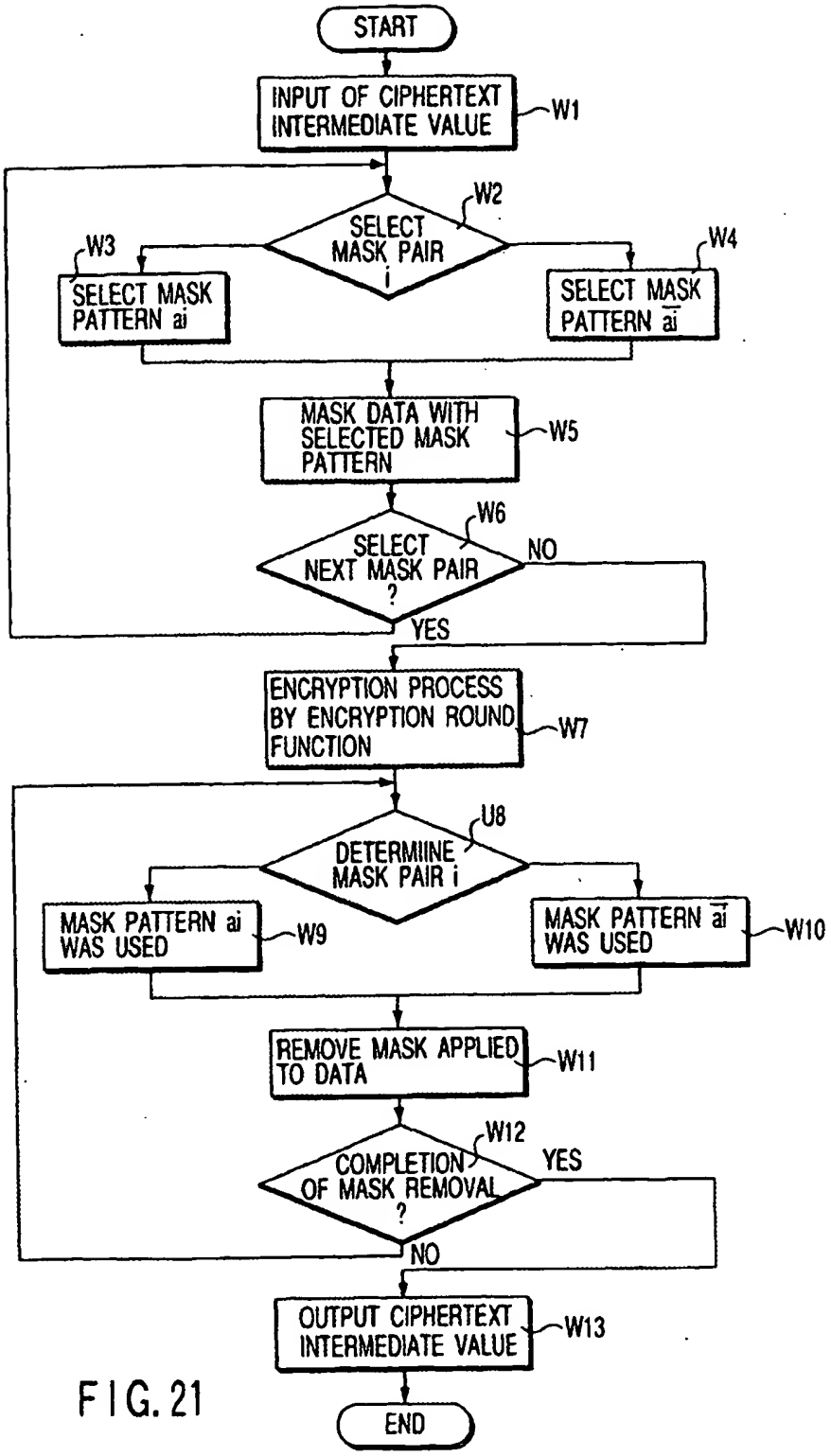

START
INPUT OF CIPHERTEXT INTERMEDIATE VALUE
W1
SELECT MASK PAIR i
W2
W3
SELECT MASK PATTERN ai
W4
SELECT MASK PATTERN $\overline{ai}$
MASK DATA WITH SELECTED MASK PATTERN
W5
SELECT NEXT MASK PAIR ?
W6
NO
YES
ENCRYPTION PROCESS BY ENCRYPTION ROUND FUNCTION
W7
DETERMIINE MASK PAIR i
U8
MASK PATTERN ai WAS USED
W9
MASK PATTERN $\overline{ai}$ WAS USED
W10
REMOVE MASK APPLIED TO DATA
W11
COMPLETION OF MASK REMOVAL ?
W12
YES
NO
OUTPUT CIPHERTEXT INTERMEDIATE VALUE
W13
END

FIG. 21

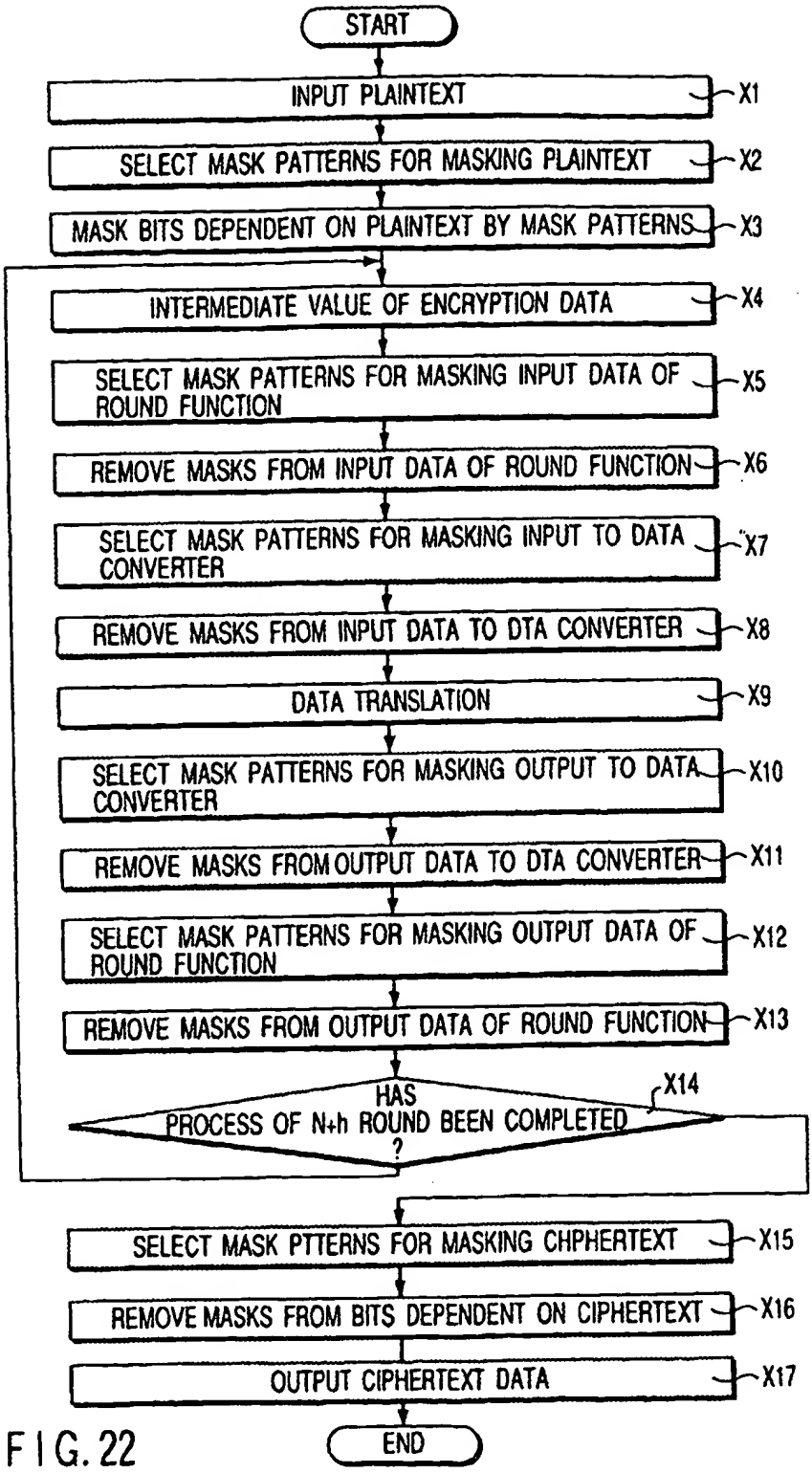

START
INPUT PLAINTEXT
X1
SELECT MASK PATTERNS FOR MASKING PLAINTEXT
X2
MASK BITS DEPENDENT ON PLAINTEXT BY MASK PATTERNS
X3
INTERMEDIATE VALUE OF ENCRYPTION DATA
X4
SELECT MASK PATTERNS FOR MASKING INPUT DATA OF ROUND FUNCTION
X5
REMOVE MASKS FROM INPUT DATA OF ROUND FUNCTION
X6
SELECT MASK PATTERNS FOR MASKING INPUT TO DATA CONVERTER
X7
REMOVE MASKS FROM INPUT DATA TO DTA CONVERTER
X8
DATA TRANSLATION
X9
SELECT MASK PATTERNS FOR MASKING OUTPUT TO DATA CONVERTER
X10
REMOVE MASKS FROM OUTPUT DATA TO DTA CONVERTER
X11
SELECT MASK PATTERNS FOR MASKING OUTPUT DATA OF ROUND FUNCTION
X12
REMOVE MASKS FROM OUTPUT DATA OF ROUND FUNCTION
X13
HAS PROCESS OF N+h ROUND BEEN COMPLETED ?
X14
SELECT MASK PTTERNS FOR MASKING CHPHERTEXT
X15
REMOVE MASKS FROM BITS DEPENDENT ON CIPHERTEXT
X16
OUTPUT CIPHERTEXT DATA
X17
END

FIG. 22

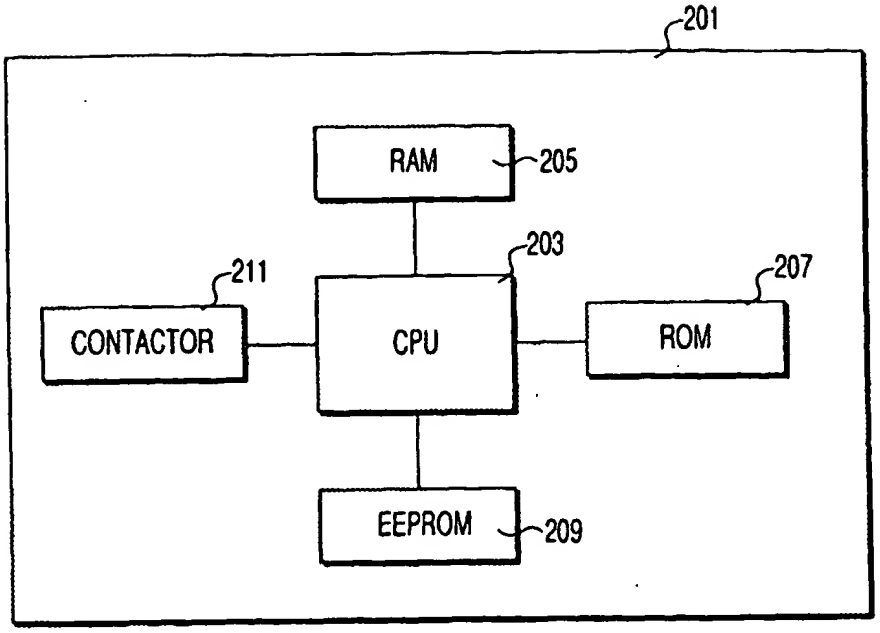


FIG. 23